

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re Patent Application of:

Yoshihide NOMURA et al.

Application No.: Unassigned

Group Art Unit: Unassigned

Filed: April 14, 2004

Examiner: Unassigned

For: APPARATUS AND METHOD FOR BUSINESS PROCESS TRACKING AND BUSINESS
PROCESS TRACKING PROGRAM, AND RECORDING MEDIUM THEREOF

**SUBMISSION OF CERTIFIED COPY OF PRIOR FOREIGN
APPLICATION IN ACCORDANCE
WITH THE REQUIREMENTS OF 37 C.F.R. § 1.55**

Commissioner for Patents
PO Box 1450
Alexandria, VA 22313-1450

Sir:

In accordance with the provisions of 37 C.F.R. § 1.55, the applicant(s) submit(s)
herewith a certified copy of the following foreign application:

Japanese Patent Application No(s). 2003-346271

Filed: October 3, 2003

It is respectfully requested that the applicant(s) be given the benefit of the foreign filing
date(s) as evidenced by the certified papers attached hereto, in accordance with the
requirements of 35 U.S.C. § 119.

Respectfully submitted,

STAAS & HALSEY LLP

By: 

David M. Pitcher

Registration No. 25,908

Date: April 14, 2004

1201 New York Ave, N.W., Suite 700
Washington, D.C. 20005
Telephone: (202) 434-1500
Facsimile: (202) 434-1501

日 本 国 特 許 庁
JAPAN PATENT OFFICE

別紙添付の書類に記載されている事項は下記の出願書類に記載されている事項と同一であることを証明する。

This is to certify that the annexed is a true copy of the following application as filed with this Office.

出 願 年 月 日 2 0 0 3 年 1 0 月 3 日
Date of Application:

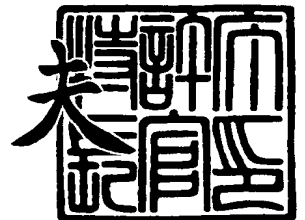
出 願 番 号 特 願 2 0 0 3 - 3 4 6 2 7 1
Application Number:
[ST. 10/C] : [J P 2 0 0 3 - 3 4 6 2 7 1]

出 願 人 富 士 通 株 式 会 社
Applicant(s):

2 0 0 4 年 2 月 2 5 日

特許庁長官
Commissioner,
Japan Patent Office

今 井 康 夫



出証番号 出証特 2 0 0 4 - 3 0 1 3 5 4 4

【書類名】 特許願
【整理番号】 0352368
【提出日】 平成15年10月 3日
【あて先】 特許庁長官殿
【国際特許分類】 G06F 17/60
【発明者】
 【住所又は居所】 神奈川県川崎市中原区上小田中 4 丁目 1 番 1 号 富士通株式会社
 内
 【氏名】 野村 佳秀
【発明者】
 【住所又は居所】 神奈川県川崎市中原区上小田中 4 丁目 1 番 1 号 富士通株式会社
 内
 【氏名】 加藤 光幾
【発明者】
 【住所又は居所】 神奈川県川崎市中原区上小田中 4 丁目 1 番 1 号 富士通株式会社
 内
 【氏名】 原 裕貴
【発明者】
 【住所又は居所】 神奈川県川崎市中原区上小田中 4 丁目 1 番 1 号 富士通株式会社
 内
 【氏名】 林 未実子
【発明者】
 【住所又は居所】 神奈川県川崎市中原区上小田中 4 丁目 1 番 1 号 富士通株式会社
 内
 【氏名】 小林 浩
【発明者】
 【住所又は居所】 神奈川県川崎市中原区上小田中 4 丁目 1 番 1 号 富士通株式会社
 内
 【氏名】 木村 修
【特許出願人】
 【識別番号】 000005223
 【氏名又は名称】 富士通株式会社
【代理人】
 【識別番号】 100087848
 【弁理士】
 【氏名又は名称】 小笠原 吉義
 【電話番号】 03-3807-1151
【選任した代理人】
 【識別番号】 100083297
 【弁理士】
 【氏名又は名称】 山谷 皓榮
【手数料の表示】
 【予納台帳番号】 012586
 【納付金額】 21,000円
【提出物件の目録】
 【物件名】 特許請求の範囲 1
 【物件名】 明細書 1
 【物件名】 図面 1
 【物件名】 要約書 1
 【包括委任状番号】 0203885

【書類名】 特許請求の範囲**【請求項 1】**

異なる業務システム間を跨がって実行される業務の流れである業務プロセスを追跡する業務プロセストラッキング装置であって、

各業務システム内の業務で扱う業務データと、各業務データ間の関連情報を定義した業務プロセス定義情報を記憶する手段と、

前記業務システム内において抽出された各業務を実行するアプリケーションの実行状態を示す情報であるイベントデータを収集するイベントデータ収集手段と、

前記収集されたイベントデータを前記業務プロセス定義情報に基づき業務データ単位でまとめ、イベントデータとして取得した業務データ間の関連付けを行うイベント関連付け手段と、

前記関連付けられた業務データを蓄積するイベント管理データ記憶手段と、

入力された検索条件に基づいて前記イベント管理データ記憶手段を検索し、業務データ間の関連をツリー形式で出力する出力手段とを備える

ことを特徴とする業務プロセストラッキング装置。

【請求項 2】

請求項 1 に記載の業務プロセストラッキング装置において、

前記出力手段は、前記出力された業務データのツリー上で選択された業務データに関連するイベントを、前記業務プロセス定義情報に基づいて作成される業務プロセスの図上に表示する手段を備える

ことを特徴とする業務プロセストラッキング装置。

【請求項 3】

異なる業務システム間を跨がって実行される業務の流れである業務プロセスを追跡する業務プロセストラッキング方法であって、

各業務システム内の業務で扱う業務データと、各業務データ間の関連情報を定義した業務プロセス定義情報を記憶するステップと、

前記業務システム内において抽出された各業務を実行するアプリケーションの実行状態を示す情報であるイベントデータを収集するステップと、

前記収集されたイベントデータを前記業務プロセス定義情報に基づき業務データ単位でまとめ、イベントデータとして取得した業務データ間の関連付けを行うステップと、

前記関連付けられた業務データをイベント管理データ記憶手段に蓄積するステップと、
入力された検索条件に基づいて前記イベント管理データ記憶手段を検索し、業務データ間の関連をツリー形式で出力するステップとを有する

ことを特徴とする業務プロセストラッキング方法。

【請求項 4】

異なる業務システム間を跨がって実行される業務の流れである業務プロセスを追跡する業務プロセストラッキング方法をコンピュータに実行させるためのプログラムであって、

各業務システム内の業務で扱う業務データと、各業務データ間の関連情報を定義した業務プロセス定義情報を記憶する処理と、

前記業務システム内において抽出された各業務を実行するアプリケーションの実行状態を示す情報であるイベントデータを収集する処理と、

前記収集されたイベントデータを前記業務プロセス定義情報に基づき業務データ単位でまとめ、イベントデータとして取得した業務データ間の関連付けを行う処理と、

前記関連付けられた業務データをイベント管理データ記憶手段に蓄積する処理と、
入力された検索条件に基づいて前記イベント管理データ記憶手段を検索し、業務データ間の関連をツリー形式で出力する処理とを、

コンピュータに実行させるための業務プロセストラッキングプログラム。

【請求項 5】

異なる業務システム間を跨がって実行される業務の流れである業務プロセスを追跡する業務プロセストラッキング方法をコンピュータに実行させるためのプログラムを記録した

記録媒体であって、

各業務システム内の業務で扱う業務データと、各業務データ間の関連情報を定義した業務プロセス定義情報を記憶する処理と、

前記業務システム内において抽出された各業務を実行するアプリケーションの実行状態を示す情報であるイベントデータを収集する処理と、

前記収集されたイベントデータを前記業務プロセス定義情報に基づき業務データ単位でまとめ、イベントデータとして取得した業務データ間の関連付けを行う処理と、

前記関連付けられた業務データをイベント管理データ記憶手段に蓄積する処理と、

入力された検索条件に基づいて前記イベント管理データ記憶手段を検索し、業務データ間の関連をツリー形式で出力する処理とを、

コンピュータに実行させるための業務プロセストラッキングプログラムを記録した記録媒体。

【書類名】明細書

【発明の名称】業務プロセストラッキング装置、業務プロセストラッキング方法、業務プロセストラッキングプログラム、業務プロセストラッキングプログラムを記録した記録媒体

【技術分野】**【0001】**

本発明は、複数の業務システムの間に流れる業務プロセスの追跡を行う業務プロセスの追跡技術に関し、特に、各業務システムから抽出したイベント情報を業務データ単位にまとめ、業務データを関連付けて管理し、入力された検索条件に基づいて業務プロセスをトラッキングする業務プロセストラッキング装置、業務プロセストラッキング方法、業務プロセストラッキングプログラム、業務プロセストラッキングプログラムを記録した記録媒体に関する。

【背景技術】**【0002】**

稼動中の既存業務の改善を行う場合、どの部分に問題があり、改善したらよいかを判断するために、業務システム間の業務プロセスの分析が行われる。業務システム間の業務がどのように流れていくかを見るシステムとして、ワークフローシステムがある。ワークフローシステムとは、一般に、既存のフローエンジンを備え、定義されたフローに従った順序でシステムの自動実行を行うアプリケーションを備えるシステムを意味する。

【0003】

図37は、従来の一般的なワークフローシステムの概要を示す図である。20はフロー管理サーバ、21はアプリケーションE、22はアプリケーションF、23はアプリケーションG、24はアプリケーションHである。アプリケーションE21～H24は、それぞれ異なるシステムに配置されている。200はワークフローE、201はワークフローF、202はワークフローG、203はアプリケーションE21～G23の実行履歴を管理するフロー管理サーバ20内の履歴データベース(DB)である。

【0004】

210はアプリケーションE21の状態を管理する状態DB、220はアプリケーションF22の状態を管理する状態DB、230はアプリケーションG23の状態を管理する状態DB、240はアプリケーションH24の状態を管理する状態DBである。

【0005】

また、アプリケーションE21、アプリケーションF22、アプリケーションG23は、それぞれワークフローE200、ワークフローF201、ワークフローG202から呼び出されて実行され、アプリケーションH24は、上記ワークフローのいずれかから呼び出されて実行されるような実行形式を採らないものとする。

【0006】

図37に示すワークフローシステムでは、異なるシステム間の業務について、ワークフローE200、ワークフローF201、ワークフローG202の順に流れることを定義しておいて、各ワークフローに対応するアプリケーションの実行履歴をフロー管理サーバ20で管理する。すなわち、フロー管理サーバ20は、ワークフローE200に対応するアプリケーションE21を実行し、アプリケーションE21の実行終了後にワークフローF201に対応するアプリケーションF22を実行し、アプリケーションF22の実行終了後にワークフローG202に対応するアプリケーションG23を実行する。

【0007】

また、下記の特許文献1には、異なるシステムに跨るようなワークフローや非定型業務についても、その履歴や進行状態の管理情報を参照したり、辿ったりすることを可能にするワークフロー支援システムおよび支援方法について記載されている(特許文献1参照)。

【特許文献1】特開平10-63747号公報

【発明の開示】

【発明が解決しようとする課題】**【0008】**

しかし、図37に示すワークフローシステムは、あくまで各アプリケーションを順次実行していくという構成を採り、フロー管理サーバ20において管理するのは各アプリケーションの実行履歴のみであったため、各アプリケーションの状態把握が困難である。このため、例えば、フロー管理サーバ20がアプリケーションE21中のいろいろな業務を管理するという要求に応えるためには、アプリケーションE21の内部の処理を細分化して、フロー管理サーバ20から呼び出して実行するなどの改造が必要となる。

【0009】

また、例えば、アプリケーションH24は、ワークフローE200～G202のいずれからも呼び出されて実行されるようになっていないため、フロー管理サーバ20は、アプリケーションH24の状態を把握できず、アプリケーションH24の状態を把握するためには、アプリケーションH24がワークフローから呼び出されて実行される形式に改造する必要がある。

【0010】

さらに、現行のワークフローシステムでは、図38に示すように、途中で複数に分岐するアプリケーションの関係を把握できないという問題がある。図38において、例えば、ワークフローF201はワークフローE200に関連した複数の処理をアプリケーションF22とアプリケーションF'22'に分けて実行させる。また、ワークフローG202は、ワークフローF201に関連した複数の処理を、それぞれアプリケーションG23とアプリケーションG'23'に分けて実行させる。アプリケーションF22のデータ、アプリケーションF'22'のデータは、アプリケーションG23およびアプリケーションG'23'の双方に渡されるものとする。

【0011】

従来の一般的なワークフローシステムでは、アプリケーションF22またはF'22'と、アプリケーションG23またはG'23'との関係を把握することができない。このため、途中で複数に分岐するアプリケーションの関係を把握するためには、複数プロセスに分割して別々に分析するなど、業務プロセスと分析方法の見直しが必要であった。

【0012】

また、上記特許文献1には、業務システム間の業務の流れの追跡結果をフロー形式で表示する技術については開示されていない。

【0013】

特に、既存の多くの業務システムは、必ずしもフロー管理サーバ20のようなフローエンジンのもとでワークフローに従って業務処理を遂行しているわけではなく、フロー管理サーバ20が存在しないような場合には、業務の流れの追跡を行うためには、個々の業務システムにアクセスして処理済みデータ、未処理データなどのデータを収集し、分析しなければならないため、現実的には業務の流れの追跡は不可能であった。

【0014】

本発明は、上記従来技術の問題点を解決し、異なる業務システム間を跨いで行われる複数のアプリケーションからなる業務プロセスの流れを、既存システムを変更することなく追跡できるようにすることを目的とする。

【課題を解決するための手段】**【0015】**

上記課題を解決するため、本発明は以下の処理を実行する。

(1) 異なる業務システムに配置される各アプリケーションの実行状態を示す情報であるイベントデータを、各アプリケーションに応じた方法で収集し、イベントキューにキューイングする。

【0016】

ここで、イベントとは、業務システム内で、ある業務が実行されたことを示すものであり、業務の開始、終了時間、および関連属性を含んだデータである。イベントデータは、

各業務システムに配置されたイベント抽出定義に従って、各業務システム毎のイベントデータ抽出用のアプリケーションによって抽出される。各業務システム内で、抽出されたイベント情報を共通のXML (extensible markup language) 形式に変換し、イベントデータを管理するイベント管理装置のイベントキューにキューイングする。このキューイングには、例えばJMS (java message service) 等が利用される。

(2) イベント管理装置内で、イベントキュー内にキューイングされたイベント情報について、業務データ毎にまとめ、業務データ間を関連付けてイベント管理データベース(DB)内に蓄積する。ここで、業務データとは、あるまとまった単位の業務の間で共有されるデータを意味する。

(3) 入力された検索条件 (例えば、イベント発生期間、関連属性等) に基づいて、業務データの絞り込みを行う。

(4) 絞り込まれた業務データに関連するデータをツリーで展開して表示し、任意のデータからの処理の追跡を行う。

(5) ツリーで展開された業務データに関連するイベントを検索し、このイベントに関連する業務をトラッキングビューで図示して、現在の業務の流れの実行状況を表示する。ここで、トラッキングとは、あらかじめ定義された業務システム間を跨ぐ業務全体の流れである業務プロセスのうち、どの業務が実行され、どの業務が実行されていないかを確認する手法をいう。

【0017】

すなわち、本発明は、異なる業務システム間を跨がって実行される業務の流れである業務プロセスを追跡する業務プロセストラッキング装置であって、各業務システム内の業務で扱う業務データと、各業務データ間の関連情報を定義した業務プロセス定義情報を記憶する手段と、前記業務システム内において抽出された各業務を実行するアプリケーションの実行状態を示す情報であるイベントデータを収集するイベントデータ収集手段と、前記収集されたイベントデータを前記業務プロセス定義情報に基づき業務データ単位でまとめ、イベントデータとして取得した業務データ間の関連付けを行うイベント関連付け手段と、前記関連付けられた業務データを蓄積するイベント管理データ記憶手段と、入力された検索条件に基づいて前記イベント管理データ記憶手段を検索し、業務データ間の関連をツリー形式で出力する出力手段とを備えることを特徴とする。

【0018】

また、前記出力手段は、前記出力された業務データのツリー上で選択された業務データに関連するイベントを、前記業務プロセス定義情報に基づいて作成される業務プロセスの図上に表示する手段を備えることを特徴とする。

【0019】

以上の各手段は、コンピュータとソフトウェアプログラムとによって実現することができ、そのプログラムをコンピュータ読み取り可能な記録媒体に記録して提供することも、ネットワークを通して提供することも可能である。

【発明の効果】

【0020】

本発明によれば、異なる業務システム間を跨いで行われる複数のアプリケーションからなる業務プロセスの流れを既存システムを変更することなく追跡することが可能となる。特に、本発明によれば、個々のアプリケーションの状態を把握することができ、また、フロー管理されないアプリケーションの状態の把握も可能となる。また、本発明によれば、複数に分岐するアプリケーション間の関係の把握が可能となる。

【発明を実施するための最良の形態】

【0021】

図1は、本発明のシステム構成例を示す図である。図1において、1はイベントデータを収集し管理するイベント管理装置、3は業務システムA、4は業務システムB、5は業務システムC、6はユーザ端末である。

【0022】

イベント管理装置 1 内において、11 はイベントデータの検索条件を入力する入力部、12 は業務システム A3 ~ 業務システム C5 から送信されたイベントデータがキューイングされるイベントキュー、13 はイベントキュー 12 内のイベントデータを業務データ単位にまとめ、業務データ間の関連付けを行うイベント関連付け部、14 は関連付けられた業務データが蓄積されるイベント管理 DB、15 は業務プロセスの定義情報と業務データの定義情報とが蓄積される業務プロセス DB である。16 はユーザ端末 6 から入力された検索条件に基づくイベント管理 DB 14 の検索結果を出力する出力部である。

【0023】

業務プロセスとは、業務システム間を跨ぐ業務全体の流れを意味する。以下では、業務プロセスが、受注、在庫照会、発注、受入、出荷、配送等の業務から構成されるものとして説明する。また、業務データとは、前述したように、あるまとまった単位の業務の間で共有されるデータを意味する。業務データは、イベント管理 DB 14 内では、各業務の伝票という形で蓄積される。

【0024】

業務システム A3 内において、30、30' は、それぞれ業務システム A3 内で業務を実行するアプリケーション A、アプリケーション A'、31 はアプリケーション A30 の状態が蓄積されている状態 DB、31' はアプリケーション A'30' の状態が蓄積されている状態 DB、32 は状態 DB 31 および状態 DB 31' からイベントデータを抽出するイベント抽出部、33 は抽出したイベントデータを、例えば XML 形式に変換するイベントデータ変換部、34 は変換されたイベントデータをイベント管理装置 1 に送信するイベントデータ送信部、320 は業務システム A3 内のどのデータ項目を抽出するかを定義したイベント抽出定義のデータ（以下、単にイベント抽出定義という）である。

【0025】

業務システム B4 内において、40、40' は業務システム B4 内で業務を実行するアプリケーション B、アプリケーション B'、41 はアプリケーション B40 の状態が蓄積されている状態 DB、41' はアプリケーション B'40' の状態が蓄積されている状態 DB、42 は状態 DB 41 または状態 DB 41' からイベントデータを抽出するイベント抽出部、43 は抽出したイベントデータを、例えば XML 形式に変換するイベントデータ変換部、44 は変換されたイベントデータをイベント管理装置 1 に送信するイベントデータ送信部、420 は業務システム B4 内のどのデータ項目を抽出するかを定義したイベント抽出定義である。

【0026】

業務システム C5 内において、50、50' は業務システム C5 内で業務を実行するアプリケーション C、アプリケーション C'、51 はアプリケーション C50 の状態が蓄積されている状態 DB、51' はアプリケーション C'50' の状態が蓄積されている状態 DB、52 は状態 DB 51 または状態 DB 51' からイベントデータを抽出するイベント抽出部、53 は抽出したイベントデータを例えば XML 形式に変換するイベントデータ変換部、54 は変換されたイベントデータをイベント管理装置 1 に送信するイベントデータ送信部、520 は業務システム C5 内のどのデータ項目を抽出するかを定義したイベント抽出定義である。

【0027】

イベント管理装置 1 のイベントキュー 12 にキューイングされるイベントデータは、例えば、イベントの種別を示すイベント種別、イベントが属しているデータの識別子である主キー、主キーに対応する業務に関連する業務のイベントが属しているデータの識別子である関連キー、製造番号、個数、単価等の属性データから構成される。

【0028】

業務システム A3 ~ 業務システム C5 は、例えば図 2 に示すように配置される。業務システム A3 は、例えば、受注と在庫照会という業務を実行し、受注伝票 35 を扱う。業務システム B4 は、例えば、発注と受入という業務を実行し、発注伝票 45 を扱う。業務システム C5 は、例えば、出荷と配送という業務を実行し、配送伝票 55 を扱う。

【0029】

本発明による業務プロセスのトラッキングの準備として、業務プロセスDB15に格納する業務プロセスおよび業務データの定義情報の入力、および各業務システムA～Cのイベント抽出部32～52に設定するイベント抽出定義320～520の入力が必要となる。これらの定義情報の入力方法について説明する。

【0030】

図3および図4は、業務プロセスDB15内に格納する定義データの入力画面を示す図である。例えば、あらかじめ用意されたツールボックス（図示省略）のグラフィカルユーザインタフェース（GUI）を用いることにより、図3の点線で囲った部分に示すような業務プロセスを描画すると、業務プロセスに関する定義データが生成され、業務プロセスDB15内に蓄積される。図3に示す入力画面では、例えばアクティビティとして「<<販売>>受注」の次に「<<販売>>在庫照会」が実行され、続いて、在庫がない場合には、「<<購買>>発注」、「<<購買>>受入」のアクティビティが実行され、在庫がある場合には、「<<販売>>出荷」、「<<配送>>配送」のアクティビティが実行されることが定義されている。

【0031】

また、図4に示す入力画面上で、「受注」、「発注」、「配送」という業務データとその間の関係を描画することにより、各業務データが生成され、業務プロセスDB15内に蓄積される。図4に示す入力画面では、例えば、業務データ「受注」の属性データとして「受注ID」や「納品日」等のデータがあり、業務データ「発注」の属性データとして「発注ID」や「受注ID」等のデータがあり、業務データ「配送」の属性データとして「配送ID」や「発注ID」等のデータがあることが定義されている。このように、業務データ内に他の業務データのIDを持つことで、業務データ間の関連付けが行われる。

【0032】

図3および図4の入力画面から入力された定義情報に基づいて、図5および図6に示すような定義データが自動生成される。この例では、業務プロセスの定義データがリレーショナルデータベースのテーブルとして生成され管理されている。図5および図6は、業務プロセスDB15内に格納する定義データのデータ構成を示す図である。

【0033】

図5（A）は業務プロセスを構成する各業務であるアクティビティのテーブルの例を示している。図5（A）に示すように、業務プロセスは、例えば、受注、在庫照会、発注、受入、出荷、配送といったアクティビティから構成されることが定義され、各アクティビティには、それらの各々を一意に特定するIDが付与される。また、図5（B）は、業務データのテーブルの例を示している。例えば、業務データのテーブルには、業務データとして受注、発注、配送があることが定義され、それぞれに各業務データを一意に特定するIDが定義されている。

【0034】

図6（A）は、業務データ内の属性データを示す図である。例えば、受注という業務データ（ID=d1）の属性データは、受注IDと納品日と受注金額であり、発注という業務データ（ID=d2）の属性データは、発注IDと受注IDと発注先であり、配送という業務データ（ID=d3）の属性データは、配送IDと発注IDと配送日であることが定義されている。

【0035】

上記のように、例えば、発注という業務データの属性データとして受注IDが定義され、また、配送という業務データの属性データとして発注IDが定義されることにより、発注は受注に対する関連を持ち、配送は発注に関連を持つこととなる。これは、上記図4において描画された「配送」→「発注」→「受注」という各業務データ間の関係に対応するものである。

【0036】

図6（B）は、アクティビティと業務データとの対応表を示す図である。この対応表は

、ID、入出力、効果、アクティビティ、業務データという項目から構成される。効果の項目の「C」はCreate、「R」はRead、「U」はUpdateを意味する。

【0037】

例えば、この対応表の第1行目を見ると、受注というアクティビティから受注という業務データが出力され、その遷移のIDはt1であることが分かる。第2行目では、在庫照会というアクティビティに受注という業務データが入力され、その遷移のIDはt2であることが分かる。また、第3行目では、発注というアクティビティから発注という業務データが出力され、その遷移のIDはt3であることが分かる。第4行目では、受入というアクティビティに発注という業務データが入力され、その遷移のIDはt4であることが分かる。第5行目では、出荷というアクティビティから配送という業務データが出力され、その遷移のIDはt5であることが分かる。第6行目では、配送というアクティビティに配送という業務データが入力され、その遷移のIDはt6であることが分かる。

【0038】

図3～図6で説明した業務プロセスの定義の後、各業務システムでどのようなイベントを監視し抽出するかイベント抽出定義を入力する。図7および図8は、各業務システム内のイベント抽出定義を生成するためのイベント抽出定義画面の一例を示す図である。図7(A)は業務プロセスの選択画面を示す。

【0039】

図7(A)においてプロセス項目として製造プロセスが選択され、「次へ」のボタンが押されると、図7(B)に示すように、アクティビティの選択画面が表示される。このアクティビティの選択画面において、例えばアクティビティとして「<<販売>>受注」が選択され、「次へ」のボタンが押されると、図8(A)に示すように、「<<販売>>受注」というアクティビティを実行する業務システムの選択画面が表示される。

【0040】

業務システムとして「業務システムA」が選択され、「次へ」のボタンが押されると、図8(B)に示すように、選択されたアクティビティのデータ項目に対応する業務システムA3内のデータ項目の選択画面が表示される。図8(B)に示す選択画面に従ってシステム内データ項目が選択され、「設定」のボタンが押されると、例えば、図9に示すようなイベント抽出定義320が生成される。図9に示すイベント抽出定義320には、例えば、選択された業務システムA3のシステム名(system name)として“myR3”が、アクティビティ名(activity Name)として“受注”が、主キー名(primary key name)として“受注”が記述され、図8(B)に示す選択画面に従って選択された業務システムA3内のデータ項目である“受注金額”、“納品希望日”が、属性名(property name)として記述される。

【0041】

図10は、イベント抽出定義の配置例を示す図である。図10に示すように、例えば、業務システムA3のイベント抽出部32は、受注抽出定義320-1と在庫照会抽出定義320-2とを有し、業務システムB4のイベント抽出部42は、発注抽出定義420-1と受入抽出定義420-2とを有し、業務システムC5のイベント抽出部52は、出荷抽出定義520-1と配送抽出定義520-2とを有する。

【0042】

業務システムA3のイベント抽出部32は、受注抽出定義320-1に従って、業務システムA3内の受注業務を実行するアプリケーションA30の状態を管理する状態DB31からイベントデータを抽出する。抽出されるイベントデータのイベント種別は受注であり、主キーは受注伝票35のIDである。

【0043】

また、在庫照会抽出定義320-2に従って、業務システムA3内の在庫照会業務を実行するアプリケーションA'30'の状態を管理する状態DB31'からイベントデータを抽出する。抽出されるイベントデータのイベント種別は在庫照会であり、主キーは受注伝票35のIDである。

【0044】

業務システムB4のイベント抽出部42は、発注抽出定義420-1に従って、業務システムB4内の発注業務を実行するアプリケーションB40の状態を管理する状態DB41からイベントデータを抽出する。抽出されるイベントデータのイベント種別は発注であり、主キーは発注伝票45のID、関連キーは業務システムA3が扱う受注伝票35のIDである。

【0045】

また、受入抽出定義420-2に従って、業務システムB4内の受入業務を実行するアプリケーションB'40'の状態を管理する状態DB41'からイベントデータを抽出する。抽出されるイベントデータのイベント種別は受入であり、主キーは発注伝票45のIDである。

【0046】

業務システムC5のイベント抽出部52は、出荷抽出定義520-1に従って、業務システムC5内の出荷業務を実行するアプリケーションC50の状態を管理する状態DB51からイベントデータを抽出する。抽出されるイベントデータのイベント種別は出荷であり、主キーは配送伝票55のID、関連キーは業務システムB4が扱う発注伝票45のIDである。

【0047】

また、配送抽出定義520-2に従って、業務システムC5内の配送業務を実行するアプリケーションC'50'の状態を管理する状態DB51'からイベントデータを抽出する。抽出されるイベントデータのイベント種別は配送であり、主キーは配送伝票55のIDである。

【0048】

イベント抽出部32～52の実現方法としては、業務システムのアプリケーションに応じて各種の方法を用いることができる。例えばSAP（service access point）／R3用イベントデータの抽出定義を用意することで、ユーザ出口ルーチンを自動的に生成し、そのモジュールを用いてイベントの抽出を行う。また、アプリケーションがリレーショナルデータベース（RDB）を用いているような場合には、RDB用イベントデータの抽出定義を用意することで、RDBのトリガモジュールを自動的に生成し、それによってイベントの抽出を行うようなこともできる。

【0049】

業務システムA3～業務システムC5において抽出された各イベントデータは、各業務システム内のイベントデータ変換部33、43、53によって、例えば図11（A）に示す受注イベントデータや図11（B）に示す発注イベントデータのように、XML形式に変換され、イベントデータ送信部34、44、54によってイベント管理装置1に送信される。

【0050】

図1に示すイベント管理装置1のイベントキュー12には、業務システムA3～業務システムC5から、各業務システム内で業務を実行する各アプリケーションに応じた方法で抽出され送信されたイベントデータがキューイングされる。

【0051】

イベント管理装置1のイベント関連付け部13は、イベントキュー12中のイベントデータを業務データ単位でまとめ、業務データ間の関連付けを行い、イベント管理DB14に蓄積する。例えば、図12に示すように、業務データの関連付けの結果、業務データAと業務データA'に関連するデータは業務データBと業務データB'であり、業務データBに関連するデータは業務データCであり、業務データB'に関連するデータは業務データC'となる。

【0052】

出力部16は、詳しくは後述するように、ユーザ端末6から入力された検索条件（例えば、業務プロセス名や伝票番号等）に基づくイベント管理DB14の検索結果を出力する

。ユーザ端末 6 は、イベント管理 DB 1 4 の検索結果を解釈し表示するイベントエクスプローラを備えており、ユーザは、このイベントエクスプローラを用いてイベント管理装置 1 にアクセスし、ユーザ端末 6 上に検索結果を表示させることができるようになっている。また、出力部 1 6 は、上記検索結果として表示された業務データのうち、選択された業務データに関連するイベントをユーザ端末 6 上にフロー形式で表示して、業務システム間に流れている業務の実行状況を表示する機能を持つ。

【0053】

ユーザ端末 6 は、ネットワークを介して http プロトコルを用いた Web アクセスによりイベント管理装置 1 にアクセスする機能を持つ。ユーザ端末 6 は、1 台に限らず何台あってもよい。

【0054】

図 1 3 は、業務プロセス DB 1 5 内のデータから生成するフロー表示用 SVG (scalable vector graphic) ファイルの例である。SVG は、画像を XML で表現するための標準的な言語であり、SVG ファイルには、この言語を用いて図 1 3 に示すように 2 次元グラフィックスを記述したデータが格納されている。この SVG ファイルは、ベクターデータ内に識別 ID を埋め込んだ形になっている。

【0055】

図 1 4 は、イベント管理装置 1 における処理フローの概要を示す図である。まず、各業務システムから受信したイベントデータをイベントキュー 1 2 に蓄積する (ステップ S 1)。次に、イベント関連付け部 1 3 が、イベントデータを業務データ単位にまとめ、業務データの関連付け処理を行う (ステップ S 2)。そして、関連付けられた業務データをイベント管理 DB 1 4 に蓄積する (ステップ S 3)。

【0056】

ユーザ端末 6 から検索条件が入力されると、出力部 1 6 は、ユーザ端末 6 から入力された検索条件に基づいてイベント管理 DB 1 4 を検索し、その検索結果を出力する (ステップ S 4)。上記検索結果として表示された業務データのうち、ある業務データをユーザが選択すると、選択された業務データに関連するイベントをユーザ端末 6 上に表示する (ステップ S 5)。表示終了の指示があれば、処理を終了する。

【0057】

図 1 5 は、業務データの関連付け処理フローを示す図である。イベント関連付け部 1 3 は、イベントデータのキューイングを検知すると (ステップ S 2 1)、イベント管理 DB 1 4 内に主キーに対応する業務データ (主業務データという) が存在するかを判断し (ステップ S 2 2)、主業務データが存在しない場合は主業務データを作成し (ステップ S 2 3)、ステップ S 2 4 に移行する。

【0058】

例えば、図 1 6 (A) に示す XML 形式のイベントデータ「イベント名：受注、主キー (発注伝票 ID) : p 1 0 1, 関連キー (受注伝票 ID) : o 0 0 1, 開始時刻 : 2 0 0 3 年 8 月 2 7 日 1 0 : 0 0, 属性 (単価) : 1 0 0 0」を受信すると、イベント管理 DB 1 4 内に「主キー : p 1 0 1」に対応する主業務データがあるかを判断し、なければ主業務データとして発注伝票を作成する。例えば、図 1 6 (B) に示すような発注伝票が作成される。

【0059】

イベント管理 DB 1 4 内に主業務データが存在する場合には、主業務データにイベントとイベントからとった属性を追加する (ステップ S 2 4)。例えば、図 1 6 (B) の発注伝票中の「<bpm:eventName> 発注</bpm:eventName>」, 「<bpm:startTime>20030827T1000</bpm:startTime>」のように、発注伝票にイベントとして「発注」というイベント名と、「2 0 0 3 年 8 月 2 7 日 1 0 : 0 0」という開始時刻を追加する。また、発注伝票中の「<bpm:property name="単価">1000</bpm:property>」のように、属性として「1 0 0 0」という単価を追加する。

【0060】

次に、イベントデータに関連キーがあるかを判断する（ステップS25）。関連キーがあれば、イベント管理DB14内に関連キーに対応する業務データ（関連業務データという）が存在するかを判断する（ステップS26）。

【0061】

イベント管理DB14内に関連業務データが存在しない場合には、関連業務データを作成し（ステップS27）、ステップS28に移行する。例えば、図16（C）に示すように、関連業務データとして受注伝票が作成される。

【0062】

関連業務データが存在する場合には、主業務データから関連業務データへの関連を追加する（ステップS28）。例えば、図16（B）に示す発注伝票中の「<bpm:relativeKey type="受注伝票">o001</bpm:relativeKey>」のように、発注伝票中に受注伝票IDの「o001」が記述され、受注伝票への関連が追加される。

【0063】

また、関連業務データから主業務データへの関連を追加する（ステップS29）。例えば、図16（C）に示す受注伝票中の「<bpm:relativeKey type="発注伝票">p001</bpm:relativeKey>」のように、受注伝票中に発注伝票IDの「p101」が記述され、発注伝票への関連が追加される。

【0064】

図17は、業務データの関連付け処理を説明する模式図である。イベント管理装置1のイベントキュー12中にイベントデータがキューイングされる。キューイングされたイベントデータは、例えば、イベント種別（またはイベント名）、主キーデータ種別、主キー、関連キーデータ種別、関連キー、関連属性、発生時刻といった要素から構成される。

【0065】

イベント関連付け部13が、上記図15で説明した業務データの関連付け処理フローに従って業務データの関連付けを行う。その結果、イベント管理DB14内に図17に示すような業務データが作成される。図17の点線で示す矢印は、イベントデータ中の各要素とイベント管理DB14中に作成される業務データとの対応を示す。

【0066】

図17において、イベント関連付け部13は、主データ項目をたててデータ種別、主キーを格納することにより、主キーに対応する主業務データを作成する。そして、主業務データ中にイベントとしてイベント種別と時刻を追加し、また、関連属性を追加する。イベントデータ中に関連キーがあれば、関連データ項目を立ててデータ種別と関連キーを格納して関連業務データを作成する。そして、主業務データ中に、リンク先情報としてデータ種別、関連キーを格納して、関連業務データへの関連を追加する。また、関連業務データ中にも、リンク先情報としてデータ種別、主キーを格納して、主業務データへの関連を追加する。

【0067】

図17に示すように、イベントデータとして取得する業務データ間の関連をもとに、あらかじめ業務データ間に双方向参照を作成することにより、後述する出力部16による検索処理において、トラッキング速度を改善することができる。

【0068】

図18は、受注イベントデータと在庫照会イベントデータと発注イベントデータを受信した場合の業務データの関連付けの例を示している。上記図15で説明した業務データの関連付け処理の結果、イベント管理DB14中に図18に示すような受注伝票と発注伝票が作成される。この受注伝票には、イベントとして「受注」と「在庫照会」が格納され、また、関連キーとして発注伝票IDの「p101」が記述され、発注伝票への関連が追加されている。また、発注伝票にも関連キーとして受注伝票IDの「o001」が記述され、受注伝票への関連が追加されている。

【0069】

図19は、初めにイベント管理DB14内にデータがない場合の業務データ（伝票）の

作成処理の例を示す図である。図19 (A) に示すように、イベント管理DB14内にデータがない場合に、イベントデータAとイベントデータBがイベントキュー12にキューイングされたとすると、イベント関連付け部13は、イベントデータAとイベントデータBとに基づいて図19 (B) に示すような受注伝票を作成する。

【0070】

すなわち、イベントデータA中の主キー「0001」に対応する主業務データとして受注伝票を作成し、イベントとして、「イベント名：受注入力」，「開始時刻：2003年8月27日11：00」，「終了時刻：2003年8月27日12：00」を追加し，「属性名：製品」と「属性：abc」，および「属性名：数量」と「属性：10」を追加する。

【0071】

図20は、イベント管理DB14内に既に図20 (A) に示す受注伝票が作成されている場合の業務データ（伝票）の作成処理の例を示す図である。イベント管理DB14内に図20 (A) に示すような受注伝票が作成されている場合に、イベントデータCとイベントデータDがイベントキュー12にキューイングされると、イベント関連付け部13は、イベントデータCとイベントデータDとに基づいて、図20 (B) に示すような受注伝票を作成する。

【0072】

すなわち、イベント関連付け部13は、図20 (A) に示す受注伝票中に、イベントとして、「イベント名：在庫照会」，「開始時刻：2003年8月27日13：00」，「終了時刻：2003年8月27日14：00」を追加して、図20 (B) に示す新たな受注伝票を作成する。

【0073】

図21は、イベント管理DB14内に図21 (A) に示すような受注伝票が作成されている場合の業務データ（伝票）の作成処理の例を示す図である。イベント管理DB14内に図21 (A) に示すような受注伝票が作成されている場合に、イベントデータEとイベントデータFがイベントキュー12にキューイングされると、イベント関連付け部13は、イベントデータEとイベントデータFとに基づいて図21 (B) に示すような受注伝票と配送伝票を作成する。

【0074】

すなわち、イベント関連付け部13は、イベントデータE中の主キーの「d001」に対応する主業務データとして配送伝票を作成し、イベントとして、「イベント名：出荷」，「開始時刻：2003年8月27日17：00」，「終了時刻：2003年8月27日18：00」を追加し，「属性名：地域」，「属性：川崎」を追加する。また、作成した配送伝票中に，「<bpm:relativeKey type="受注">o001</bpm:relativeKey>」を記述することにより、受注伝票への関連を追加するとともに、受注伝票中に「<bpm:relativeKey type="配送">d001</bpm:relativeKey>」を記述することにより、配送伝票への関連を追加する。

【0075】

図22ないし図27は、複数の業務システムからのイベント収集の流れを説明する図である。図22に示すように、例えば、業務システムA3において斜線で示す受注業務が終了した際にイベント抽出部32が抽出したイベントデータ「イベント名：受注、主伝票：受注、主キー：0001、開始時刻：10：00、終了時刻：10：30」が、イベント管理装置1のイベントキュー12内にキューイングされると、イベント管理DB14の中に、斜線で示すように、主データ項目として「受注：0001」が記述されて受注伝票が作成される。また、作成された受注伝票に、斜線で示すように、「イベント名：受注」，「開始時刻：10：00、終了時刻：10：30」というイベントが追加される。

【0076】

図23に示すように、業務システムA3において斜線で示す在庫照会業務が終了した際に、イベント抽出部32が抽出したイベントデータ「イベント名：在庫照会、主伝票：受

注、主キー：o001、開始時刻：10:30、終了時刻：11:00」が、イベントキュー12内にキューイングされると、このイベントデータの内容に基づいて、斜線で示すように、既に作成されている受注伝票の中に「イベント名：在庫照会、開始時刻：10:30、終了時刻：11:00」というイベントが追加される。

【0077】

また、図24に示すように、業務システムB4において斜線で示す発注業務が終了した際にイベント抽出部42が抽出したイベントデータ「イベント名：発注、主伝票：発注、主キー：p101、関連伝票：受注、関連キー：o001、開始時刻：11:00、終了時刻：11:30」が、イベントキュー12内にキューイングされると、以下のようにして業務データ（伝票）の関連付けが行われる。

【0078】

まず、主データ項目として斜線で示す「発注：p101」をたてて、主キー「p101」に対応する発注伝票を作成する。そして、作成した発注伝票中に、「イベント名：発注、開始時刻：11:00、終了時刻：11:30」というイベントを追加する。イベントデータ中には、関連キーとして受注伝票IDの「o001」が記述されているので、発注伝票の中に斜線で示す受注伝票IDの「o001」を記述し、受注伝票との関連付けを行う。また、受注伝票の中に斜線で示す発注伝票IDの「p101」を記述し、受注伝票へとの関連付けを行う。

【0079】

また、図25に示すように、業務システムB4において斜線で示す受入業務が終了した際に、イベント抽出部42が抽出したイベントデータ「イベント名：受入、主伝票：発注、主キー：p101、開始時刻：11:30、終了時刻：12:00」が、イベントキュー12内にキューイングされると、このイベントデータの内容に基づいて、既に作成されている発注伝票の中に斜線で示す「イベント名：受入、開始時刻：11:30、終了時刻：12:00」というイベントが追加される。

【0080】

また、図26に示すように、業務システムC5において斜線で示す出荷業務が終了した際に、イベント抽出部52が抽出したイベントデータ「イベント名：出荷、主伝票：配送、主キー：d201、関連伝票：発注、関連キー：p101、開始時刻：12:00、終了時刻：12:30」が、イベントキュー12内にキューイングされると、以下のようにして業務データ（伝票）の関連付けが行われる。

【0081】

まず、主データ項目として「配送：d201」をたてて、主キー「d201」に対応する配送伝票を作成する。そして、「イベント名：出荷、開始時刻：12:00、終了時刻：12:30」というイベントが追加される。さらに、イベントデータ中には、関連キーとして発注伝票IDの「p101」が定義されているので、配送伝票の中に斜線で示す発注伝票IDの「p101」を記述し、発注伝票との関連付けを行う。また、発注伝票の中に斜線で示す配送伝票IDの「d201」を記述し、配送伝票へとの関連付けを行う。

【0082】

また、図27に示すように、業務システムC5から斜線で示す配送業務が終了した際に、イベント抽出部52が抽出したイベントデータ「イベント名：配送、主伝票：配送、主キー：d201、開始時刻：12:30、終了時刻：13:00」が、イベントキュー12内にキューイングされると、このイベントデータの内容に基づいて、既に作成されている配送伝票の中に斜線で示す「イベント名：配送、開始時刻：12:30、終了時刻：13:00」というイベントが追加される。

【0083】

図28および図29は、ユーザ端末6における業務データの検索条件入力画面の例を示す図である。ユーザは、図28に示す検索条件入力画面を利用して、どのデータに注目して業務の流れをトラッキングするかについての条件をイベント管理装置1に入力する。

【0084】

図28に示す検索条件入力画面の例では、業務プロセスとして「sample1」、検索対象として「配送」伝票が選択され、検索項目として「伝票番号」、検索範囲として「2003年1月1日～2003年10月1日」が選択されている。また、結果表示方法として、「イベントエクスプローラで表示」が選択されている。「検索実行」が押されると、イベント管理装置1内のイベント管理DB14の検索が開始される。検索結果の表示例については、図31、図32、図33を用いて説明する。

【0085】

図29は、検索条件入力画面において検索結果の表示方法として「一覧で表示」を選択した場合の例を示す図である。この例では、業務プロセスとして「製造販売」、検索対象として「受注」、検索範囲として「2003年1月1日～2003年1月1日」が選択されている。さらに、結果表示方法として「一覧で表示」が選択され、「検索実行」が押されると、例えば図30に示すような検索結果が表示される。この例では、検索結果として伝票番号「0002、0001、0011」が表示されている。

【0086】

図31は、業務プロセスのトラッキング過程における操作と表示の一例を示す図である。ここでは、検索条件入力画面において選択された業務プロセスが「製品プロセス」である場合の例を示している。イベント管理DB14が検索された結果、例えば、図31(A)に示すように、ユーザ端末6のイベントエクスプローラの表示領域に受注伝票の主データ項目である「受注：0001」が表示され、その横に製品プロセスが表示される。このイベントエクスプローラの領域に表示された「+受注：0001」は、「受注：0001」というノードが開いていること、すなわち、受注伝票の主データ項目が選択されていないことを示している。

【0087】

次に、表示されたイベントエクスプローラを利用して「受注：0001」が選択されると、受注伝票の主データ項目が選択されたことから、図31(B)に示すように、「受注：0001」というノードが開いて「-受注：0001」と表示され、関連する業務データである配送伝票への関連データ「+配送：d001」が表示される。「受注：0001」が選択されたことにより、受注伝票に関連する「受注入力」のイベントと「在庫照会」のイベントが検索され、イベントエクスプローラの横に表示された製品プロセスにおいて、例えば、網かけ部に示すように、両イベントに関連するアクティビティの色が変更されて表示される。色を変えるのではなく、文字、線の太さなどの属性を変化させて表示してもよい。これは、あらかじめフローの図示に用いるデータをSVG (scalable vector graphic) で作成し、SVG内のオブジェクトの色、文字、線の太さなどの属性を変化させることで簡易に実現することができる。

【0088】

次に、図31(B)に示す「+配送：d001」が選択されると、図31(C)のイベントエクスプローラの領域に示すように、「配送：d001」というノードが開いて、「-配送：d001」という表示に変わる。そして、選択された「配送：d001」に関連する「出荷」のイベントが検索され、製品プロセスにおいて、例えば、網かけ部に示すように、「出荷」のイベントに関連するアクティビティの色が変更されて表示される。

【0089】

図32は、業務プロセスの逆トラッキング過程における操作と表示の一例を示す図である。イベント管理DB14が検索された結果、例えば、図32(A)に示すように、ユーザ端末6のイベントエクスプローラの表示領域に配送伝票の主データ項目である「配送：d001」が表示され、その横に、検索対象である製品プロセスが表示されたとする。このイベントエクスプローラの画面で「配送：d001」が選択されると、配送伝票の主データ項目が選択されたことから、図32(B)に示すように、「配送：d001」というノードが開いて「-配送：d001」と表示され、関連する業務データである受注伝票への関連データ「+受注：0001」が表示される。「配送：d001」が選択されたことにより、配送伝票に関連する「出荷」のイベントが検索され、図32(B)に示す製品プ

ロセスにおいて、例えば、網かけ部に示すように、「出荷」のイベントに関連するアクティビティの色が変更されて表示される。

【0090】

次に、図32(B)に示す「+受注：o001」が選択されると、図32(C)のイベントエクスプローラの領域に示すように、「受注：o001」というノードが開いて、「-受注：o001」という表示に変わる。そして、選択された「受注：o001」に関連する「受注入力」のイベントと「在庫照会」のイベントが検索され、図32(C)に示す製品プロセスにおいて、例えば、網かけ部に示すように、これらのイベントに関連するアクティビティの色が変更されて表示される。このようにイベントを順次さかのぼる逆トラッキングも可能である。

【0091】

図33は、業務プロセスのトラッキング結果画面の例を示す図である。ここでは、検索条件入力画面において、業務プロセスとして「sample1」、検索対象として「配送」、検索範囲として「2003年1月1日～2003年10月1日」が選択された場合の検索結果と表示される業務プロセスを示す。

【0092】

図33のイベントエクスプローラの表示領域に示すように、配送伝票の主データ項目である「配送：d201」が選択されたことから、配送伝票に関連する「出荷」のイベントと「配送」のイベントが検索され、業務プロセスのトラッキング結果として、この「出荷」のイベントと「配送」のイベントに関連するアクティビティが、網かけ部に示すように色が変更されて表示される。

【0093】

図34は、イベントエクスプローラの表示処理フローを示す図である。まず、未処理の検索結果が存在するかを判断し(ステップS31)、未処理の検索結果が存在する場合には、ユーザ端末6から入力された検索条件に基づいて、検索結果データを取得する(ステップS32)。例えば、図31(A)に表示されるイベントエクスプローラにおいて、検索結果データとして「受注：o001」が取得される。

【0094】

次に、取得された検索結果データが選択されているかを判断する(ステップS33)。検索結果データが選択されていない場合には、閉じたノードを描画し(ステップS34)ステップS31に戻る。例えば、上述した図31(A)に示す「+受注：o001」のように描画する。

【0095】

検索結果データが選択されている場合には、開いたノードを描画する(ステップS35)。例えば、図31(B)に示す「-受注：o001」のように描画する。次に、関連データが存在するかを判断し(ステップS36)、関連データが存在しなければステップS31に戻る。関連データが存在すれば関連データを取得する(ステップS37)。次に、関連データが選択されているかを判断し(ステップS38)、選択されていない場合は閉じたノードを描画して(ステップS39)、ステップS36に戻る。例えば、図31(B)において、選択された「受注：o001」の関連データとして取得された「配送：d001」を、「+配送：d001」のように描画する。

【0096】

関連データが選択されている場合には、開いたノードを描画する(ステップS40)。例えば、図31(C)に示す「-配送：d001」のように描画する。未処理の検索結果が存在しなくなった場合には、編集したビュー(全体の画面)を出力して(ステップS41)、処理を終了する。

【0097】

図35は、業務プロセスのトラッキング表示処理フローを示す図である。まず、選択済み子ノードが存在するかを判断する(ステップS51)。例えば、図33に表示されたイベントエクスプローラにおいて、「-」のついたデータが存在するかを判断する。図33

に示すイベントエクスプローラの表示例では、「配送：d201」が選択済みであることが分かる。次に、選択された子ノードのデータを特定し（ステップS52）、特定されたデータに関連するイベント情報を取得する（ステップS53）。例えば、選択された「配送：d201」に関連する「出荷」のイベント情報と「配送」のイベント情報が取得される。

【0098】

次に、未処理イベント情報が存在するかを判断する（ステップS54）。未処理イベント情報が存在する場合には、イベントに対応するアクティビティの色を変更する（ステップS55）。例えば、図33に示す業務プロセスにおいて、網かけ部に示すように、「出荷」と「配送」というイベントに関連するアクティビティの色が変更される。そして、指定された属性値を表示して（ステップS56）、ステップS53に戻る。ステップS54で未処理イベント情報が存在しない場合には、ステップS51へ戻り、選択済み子ノードが存在しなければビュー出力し（ステップS57）、処理を終了する。

【0099】

図36は、アラート表示画面の例を示す図である。ユーザ端末6からアラート表示をする対象となる業務プロセスやアラート種別等が選択入力されることにより、イベント管理装置1の出力部16は、アラート表示を出力する。

【0100】

図36に示す例では、アラート種別として全てのアラートが選択され、業務プロセスとして「sample1」が指定された場合に、斜線部に示すように、アクティビティのうち「配送」が納期遅れであることを通知するアラート表示（例えば、色の変更）がされる。アラート表示を行う条件は、あらかじめ業務プロセスDB15に業務プロセス定義情報を登録する際に定義しておくことができる。すなわち、各イベントとしての業務データの標準処理時間をあらかじめ定義しておくことで、標準処理時間に対する遅れ度、特定のイベントが完了するまでの予測時間を図示することができる。

【0101】

例えば、定義した標準処理時間に基づいて算出したイベント完了の予測時間をもとに、遅れると予想される伝票を業務プロセスの図上に表示したり、またその伝票の担当者にメールやその他の手段により通知するようなことも可能である。このように、本発明によれば、イベント管理DB14に全ての業務システムA3～C5におけるイベント情報が関連付けられて格納されているので、イベント管理DB14のデータを監視することにより、必要な場合にはアラート表示を簡易に行うことができる。

【0102】

以上から把握できるように、本発明の実施形態の特徴を述べると、以下のとおりである。

【0103】

（付記1）異なる業務システム間を跨がって実行される業務の流れである業務プロセスを追跡する業務プロセストラッキング装置であって、

各業務システム内の業務で扱う業務データと、各業務データ間の関連情報を定義した業務プロセス定義情報を記憶する手段と、

前記業務システム内において抽出された各業務を実行するアプリケーションの実行状態を示す情報であるイベントデータを収集するイベントデータ収集手段と、

前記収集されたイベントデータを前記業務プロセス定義情報に基づき業務データ単位でまとめ、イベントデータとして取得した業務データ間の関連付けを行うイベント関連付け手段と、

前記関連付けられた業務データを蓄積するイベント管理データ記憶手段と、

入力された検索条件に基づいて前記イベント管理データ記憶手段を検索し、業務データ間の関連をツリー形式で出力する出力手段とを備える

ことを特徴とする業務プロセストラッキング装置。

【0104】

(付記 2) 付記 1 に記載の業務プロセストラッキング装置において、
前記出力手段は、前記出力された業務データのツリー上で選択された業務データに関連するイベントを、前記業務プロセス定義情報に基づいて作成される業務プロセスの図上に表示する手段を備える

ことを特徴とする業務プロセストラッキング装置。

【0105】

(付記 3) 付記 2 に記載の業務プロセストラッキング装置において、
前記業務プロセス定義情報として、イベントとしての各業務データの標準処理時間の定義情報を持ち、

前記出力手段は、前記イベントの表示の際に、標準処理時間に対する遅れ度またはそのイベントの処理が完了するまでの予測時間を算出し表示する

ことを特徴とする業務プロセストラッキング装置。

【0106】

(付記 4) 付記 1、付記 2 または付記 3 に記載の業務プロセストラッキング装置において、

前記各業務システムは、前記業務プロセス定義情報に基づき、各業務システムにおいて抽出すべきイベント情報を定義したイベント抽出定義情報の記憶手段と、

あらかじめ定義されたイベント抽出定義情報に基づきイベントデータを抽出するイベント抽出手段とを備える

ことを特徴とする業務プロセストラッキング装置。

【0107】

(付記 5) 付記 1 から付記 4 までのいずれかに記載の業務プロセストラッキング装置において、

前記各業務システムは、前記イベント抽出手段が抽出したイベントデータを、各業務システムに共通のマークアップ言語で表現されるデータに変換するイベントデータ変換手段を備える

ことを特徴とする業務プロセストラッキング装置。

【0108】

(付記 6) 異なる業務システム間を跨がって実行される業務の流れである業務プロセスを追跡する業務プロセストラッキング方法であって、

各業務システム内の業務で扱う業務データと、各業務データ間の関連情報を定義した業務プロセス定義情報を記憶するステップと、

前記業務システム内において抽出された各業務を実行するアプリケーションの実行状態を示す情報であるイベントデータを収集するステップと、

前記収集されたイベントデータを前記業務プロセス定義情報に基づき業務データ単位でまとめ、イベントデータとして取得した業務データ間の関連付けを行うステップと、

前記関連付けられた業務データをイベント管理データ記憶手段に蓄積するステップと、
入力された検索条件に基づいて前記イベント管理データ記憶手段を検索し、業務データ間の関連をツリー形式で出力するステップとを有する

ことを特徴とする業務プロセストラッキング方法。

【0109】

(付記 7) 異なる業務システム間を跨がって実行される業務の流れである業務プロセスを追跡する業務プロセストラッキング方法をコンピュータに実行させるためのプログラムであって、

各業務システム内の業務で扱う業務データと、各業務データ間の関連情報を定義した業務プロセス定義情報を記憶する処理と、

前記業務システム内において抽出された各業務を実行するアプリケーションの実行状態を示す情報であるイベントデータを収集する処理と、

前記収集されたイベントデータを前記業務プロセス定義情報に基づき業務データ単位でまとめ、イベントデータとして取得した業務データ間の関連付けを行う処理と、

前記関連付けられた業務データをイベント管理データ記憶手段に蓄積する処理と、
入力された検索条件に基づいて前記イベント管理データ記憶手段を検索し、業務データ間の関連をツリー形式で出力する処理とを、
コンピュータに実行させるための業務プロセストラッキングプログラム。

【0110】

(付記8) 異なる業務システム間を跨がって実行される業務の流れである業務プロセスを追跡する業務プロセストラッキング方法をコンピュータに実行させるためのプログラムを記録した記録媒体であって、

各業務システム内の業務で扱う業務データと、各業務データ間の関連情報を定義した業務プロセス定義情報を記憶する処理と、

前記業務システム内において抽出された各業務を実行するアプリケーションの実行状態を示す情報であるイベントデータを収集する処理と、

前記収集されたイベントデータを前記業務プロセス定義情報に基づき業務データ単位でまとめ、イベントデータとして取得した業務データ間の関連付けを行う処理と、

前記関連付けられた業務データをイベント管理データ記憶手段に蓄積する処理と、

入力された検索条件に基づいて前記イベント管理データ記憶手段を検索し、業務データ間の関連をツリー形式で出力する処理とを、

コンピュータに実行させるための業務プロセストラッキングプログラムを記録した記録媒体。

【図面の簡単な説明】

【0111】

【図1】 本発明のシステム構成例を示す図である。

【図2】 業務システムの配置例を示す図である。

【図3】 業務プロセスDB内に格納する定義データの入力画面を示す図である。

【図4】 業務プロセスDB内に格納する定義データの入力画面を示す図である。

【図5】 業務プロセスDB内に格納する定義データのデータ構成を示す図である。

【図6】 業務プロセスDB内に格納する定義データのデータ構成を示す図である。

【図7】 イベント抽出定義画面の一例を示す図である。

【図8】 イベント抽出定義画面の一例を示す図である。

【図9】 イベント抽出定義の一例を示す図である。

【図10】 イベント抽出定義の配置例を示す図である。

【図11】 イベントデータの一例を示す図である。

【図12】 業務データの関連付けの結果を示す図である。

【図13】 フロー表示用SVGファイルの例を示す図である。

【図14】 イベント管理装置における処理フローの概要を示す図である。

【図15】 業務データの関連付け処理フローを示す図である。

【図16】 イベントデータまたは発注伝票、受注伝票を示す図である。

【図17】 業務データの関連付け処理を説明する模式図である。

【図18】 業務データの関連付けを示す図である。

【図19】 業務データ(伝票)の作成処理の例を示す図である。

【図20】 業務データ(伝票)の作成処理の例を示す図である。

【図21】 業務データ(伝票)の作成処理の例を示す図である。

【図22】 複数の業務システムからのイベント収集の流れを説明する図である。

【図23】 複数の業務システムからのイベント収集の流れを説明する図である。

【図24】 複数の業務システムからのイベント収集の流れを説明する図である。

【図25】 複数の業務システムからのイベント収集の流れを説明する図である。

【図26】 複数の業務システムからのイベント収集の流れを説明する図である。

【図27】 複数の業務システムからのイベント収集の流れを説明する図である。

【図28】 業務データの検索条件入力画面の例を示す図である。

【図29】 業務データの検索条件入力画面の例を示す図である。

【図 3 0】 検索結果の表示例を示す図である。

【図 3 1】 業務プロセスのトラッキング過程における操作と表示の一例を示す図である。

【図 3 2】 業務プロセスの逆トラッキング過程における操作と表示の一例を示す図である。

【図 3 3】 業務プロセスのトラッキング結果画面の例を示す図である。

【図 3 4】 イベントエクスプローラの表示処理フローを示す図である。

【図 3 5】 業務プロセスのトラッキング表示処理フローを示す図である。

【図 3 6】 アラート表示画面の例を示す図である。

【図 3 7】 従来の一般的なワークフローシステムの概要を示す図である。

【図 3 8】 従来の一般的なワークフローシステムの一例を示す図である。

【符号の説明】

【 0 1 1 2 】

1 イベント管理装置

3 ~ 5 業務システム A ~ C

6 ユーザ端末

1 1 入力部

1 2 イベントキュー

1 3 イベント関連付け部

1 4 イベント管理 DB

1 5 業務プロセス DB

1 6 出力部

2 0 フロー管理サーバ

2 1 ~ 2 4 アプリケーション E ~ H

3 0 アプリケーション A

3 1, 3 1', 4 1, 4 1', 5 1, 5 1', 2 1 0, 2 2 0, 2 3 0, 2 4 0 状態 D

B

3 2, 4 2, 5 2 イベント抽出部

3 3, 4 3, 5 3 イベントデータ変換部

3 4, 4 4, 5 4 イベントデータ送信部

4 0 アプリケーション B

5 0 アプリケーション C

2 0 0 ~ 2 0 2 ワークフロー E ~ G

2 0 3 履歴 DB

3 2 0, 4 2 0, 5 2 0 イベント抽出定義

3 2 0 - 1 受注抽出定義

3 2 0 - 2 在庫照会抽出定義

4 2 0 - 1 発注抽出定義

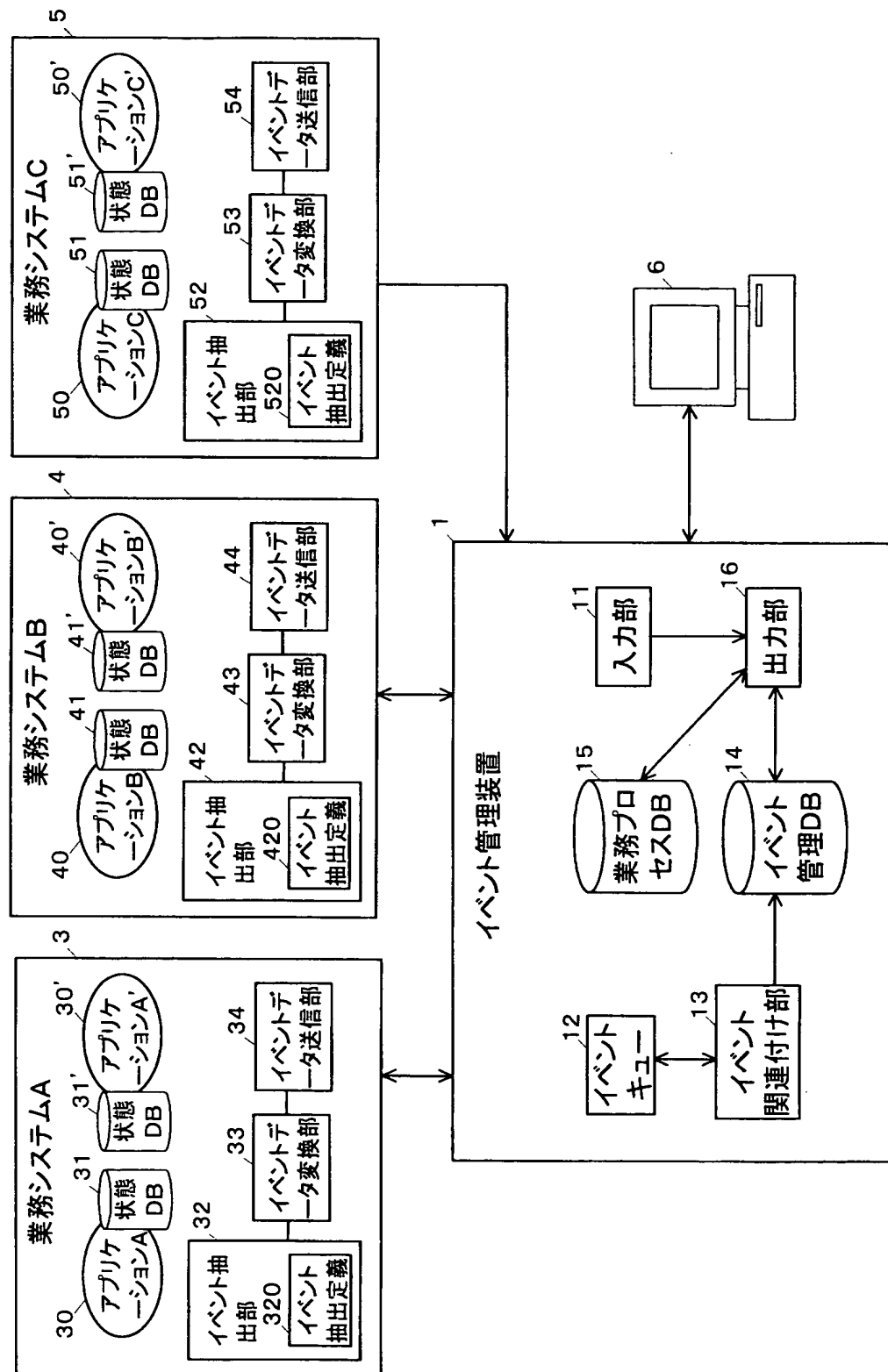
4 2 0 - 2 受入抽出定義

5 2 0 - 1 出荷抽出定義

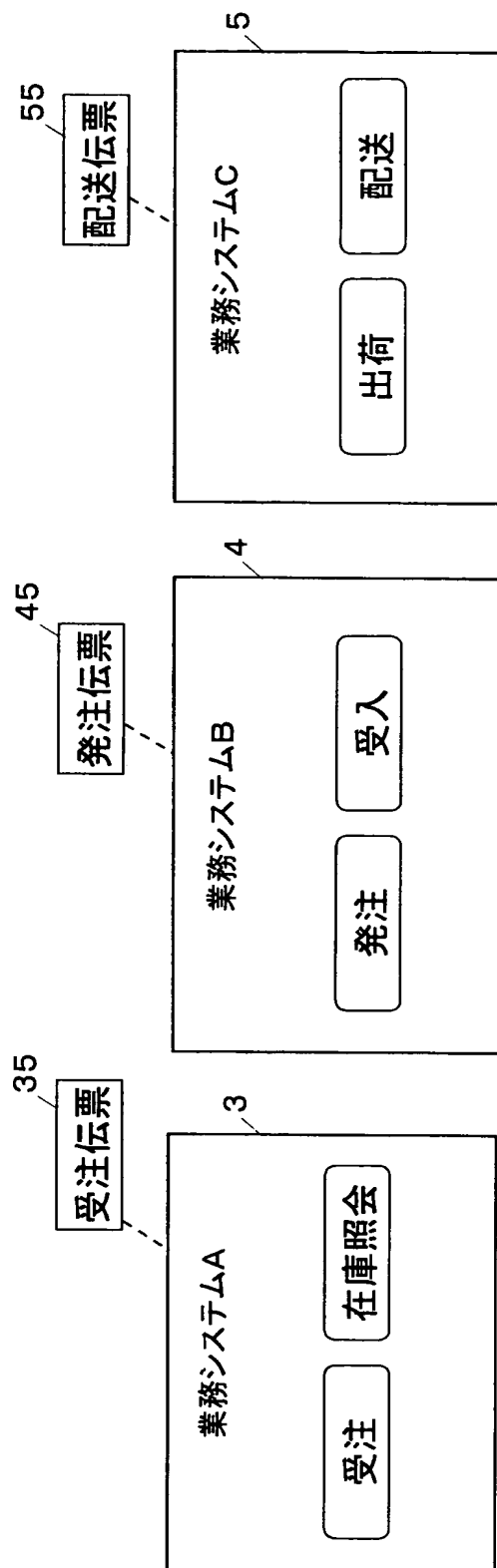
5 2 0 - 2 配送抽出定義

【書類名】 図面

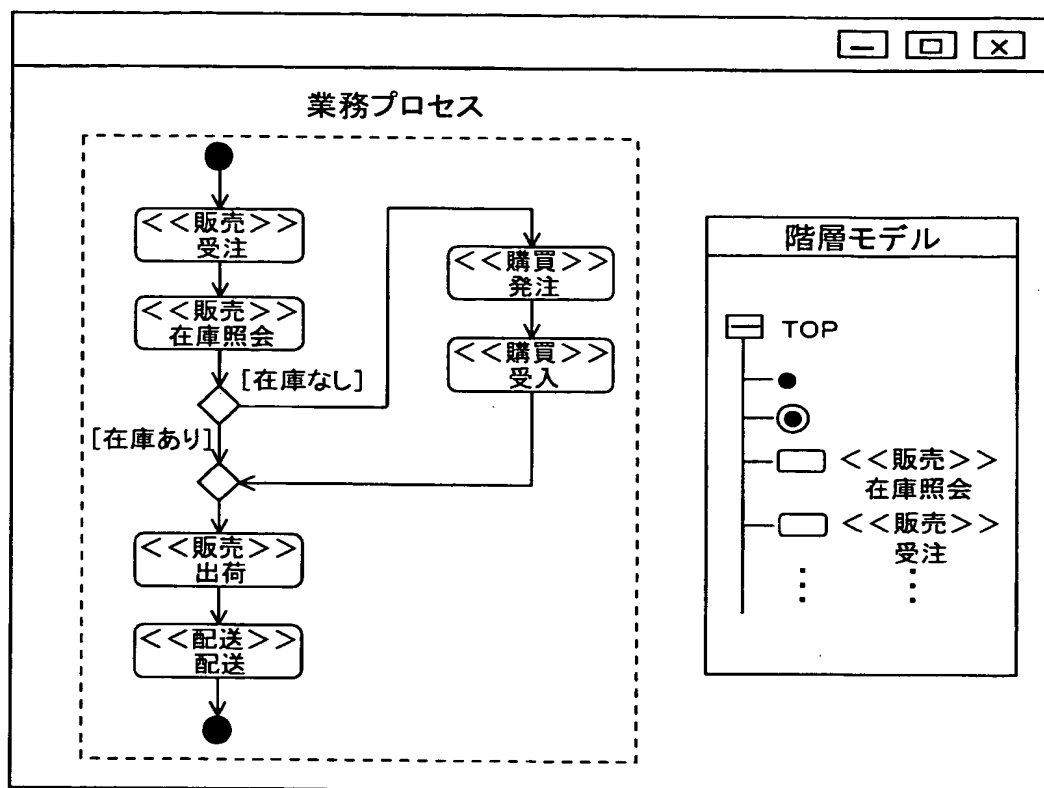
【図 1】



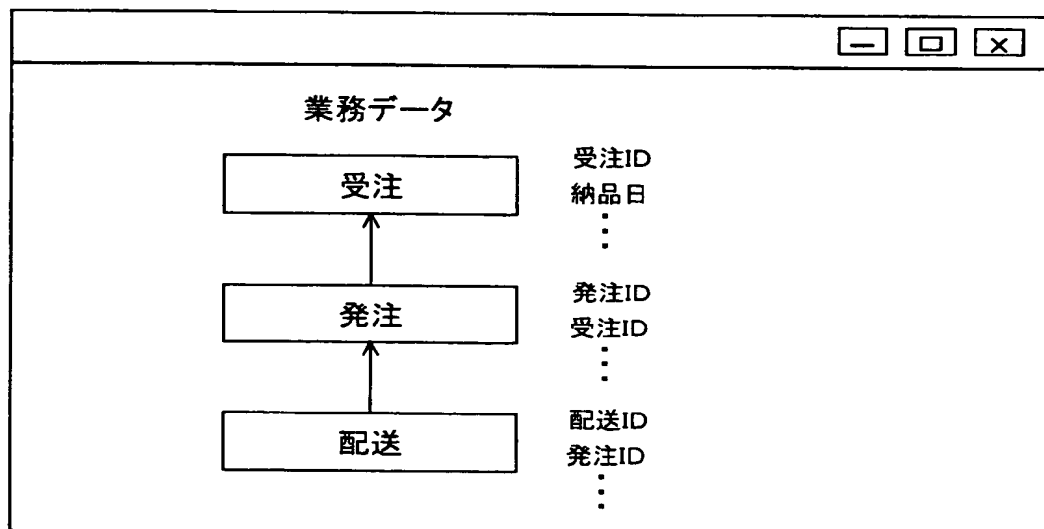
【図 2】



【図 3】



【図 4】



【図 5】

(A)

業務プロセス

ID	アクティビティ
p1	受注
p2	在庫照会
p3	発注
p4	受入
p5	出荷
p6	配送

(B)

業務データ

ID	業務データ名
d1	受注
d2	発注
d3	配送

【図 6】

(A)

データ項目

ID	名前	タイプ	業務データ
v1	受注ID	キー	d1
v2	納品日	日付	d1
v3	受注金額	数値	d1
v4	発注ID	キー	d2
v5	受注ID	キー	d2
v6	発注先	文字列	d2
v7	配送ID	キー	d3
v8	発注ID	キー	d3
v9	配送日	日付け	d3

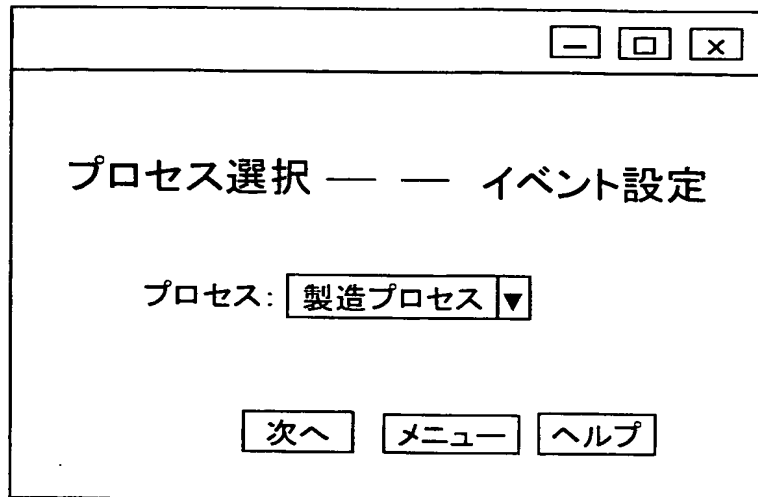
(B)

遷移

ID	入出力	効果	アクティビティ	業務データ
t1	出力	C	p1	d1
t2	入力	R	p2	d1
t3	出力	C	p3	d2
t4	入力	U	p4	d2
t5	出力	C	p5	d3
t6	入力	U	p6	d3

【図 7】

(A)

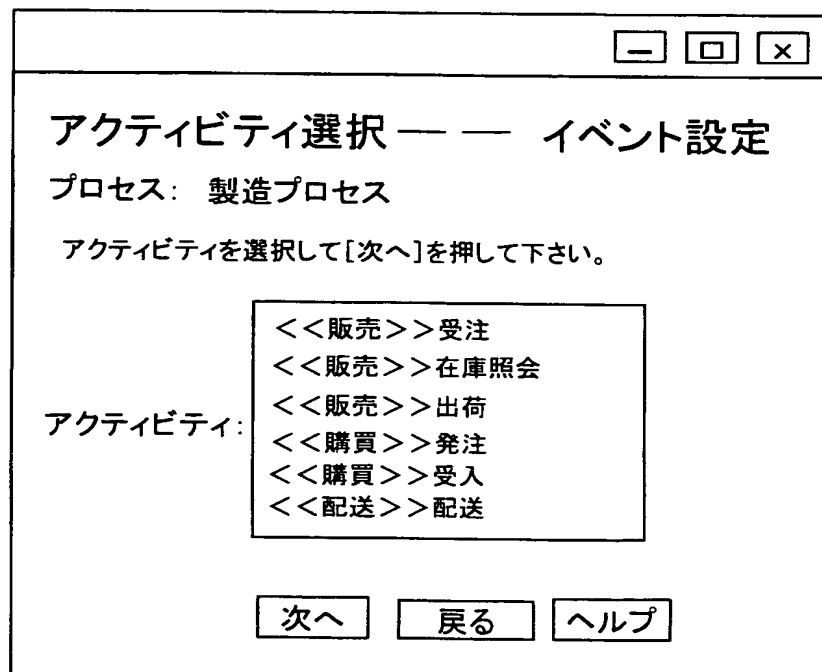


プロセス選択 — イベント設定

プロセス: 製造プロセス ▼

次へ メニュー ヘルプ

(B)



アクティビティ選択 — イベント設定

プロセス: 製造プロセス

アクティビティを選択して[次へ]を押して下さい。

アクティビティ:

- <<販売>>受注
- <<販売>>在庫照会
- <<販売>>出荷
- <<購買>>発注
- <<購買>>受入
- <<配送>>配送

次へ 戻る ヘルプ

【図 8】

(A)

システム選択 — — イベント設定

プロセス： 製造プロセス

アクティビティ： <<販売>>受注

アクティビティ「<<販売>>受注」を実行するシステムを選択してください。

業務システム：

業務システムA▼

次へ

戻る

ヘルプ

(B)

データ対応設定 — — イベント設定

プロセス： 製造プロセス

アクティビティ： <<販売>>受注

業務システム： 業務システムA

アクティビティのデータ項目に対応するシステム内データ項目を設定してください。

アクティビティデータ項目		システム内データ項目	
受注伝票	⇒	aaaa	▼
納品希望日	⇒	bbbb	▼
受注金額	⇒	cccc	▼
⋮	⋮	⋮	

設定

戻る

ヘルプ

【図 9】

イベント抽出定義

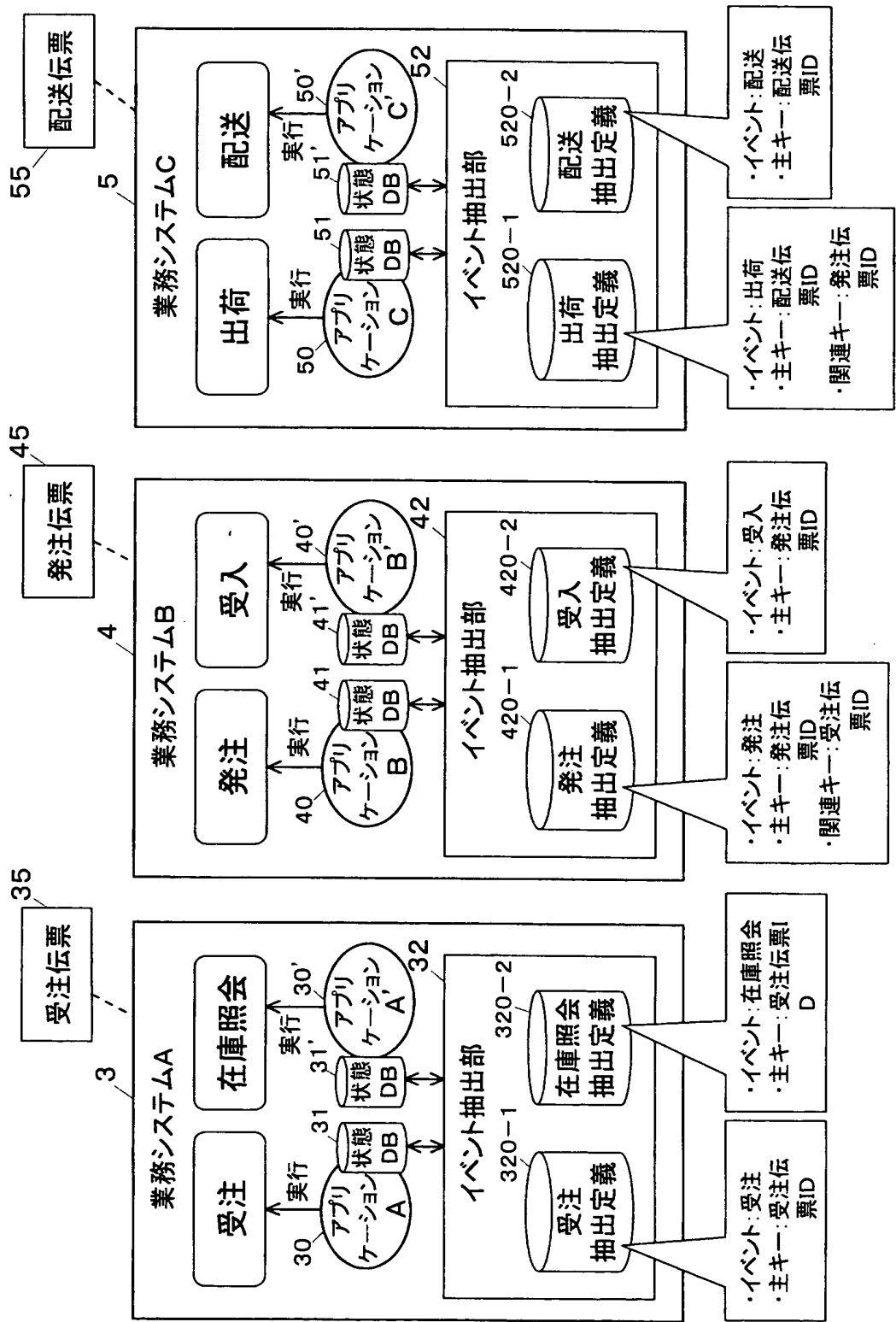
320

```

<?xml version="1.0" encoding="Shift_JIS"?>
<extractDefinition>
  <UMLmodel name="sample1">
    <system name="myR3" type="R/3">
      <location name="userexit_jutyuu_enter" activityName="受注" modelOutput="yes"> <!--アクティビティ開始-->
        <startTime /> <!-- templateのDateTimeフィールドを用いる -->
        <primaryKey name="受注" place="AA-BB" />
      </location>
      <location name="userexit_jutyuu_exit" activityName="受注" modelOutput="no"> <!--アクティビティ終了-->
        <finishTime />
        <primaryKey name="受注" place="AA-BB" />
        <property name="受注金額" place="AB-CD" />
        <property name="納品希望日" place="XX-YY" />
      </location>
      <temporaryTable> <!-- for JMS publisher -->
        <name>tempTable</name> <!-- default: "tempTable" -->
        <jdbc user="ana" password="ana" url="jdbc:oracle:thin:@localhost:1521:HESTER"
              driver="oracle.jdbc.driver.OracleDriver" />
      </temporaryTable>
    </system>
  </UMLmodel>
</extractDefinition>

```

【図 10】



【図 11】

(A)

受注イベントデータ

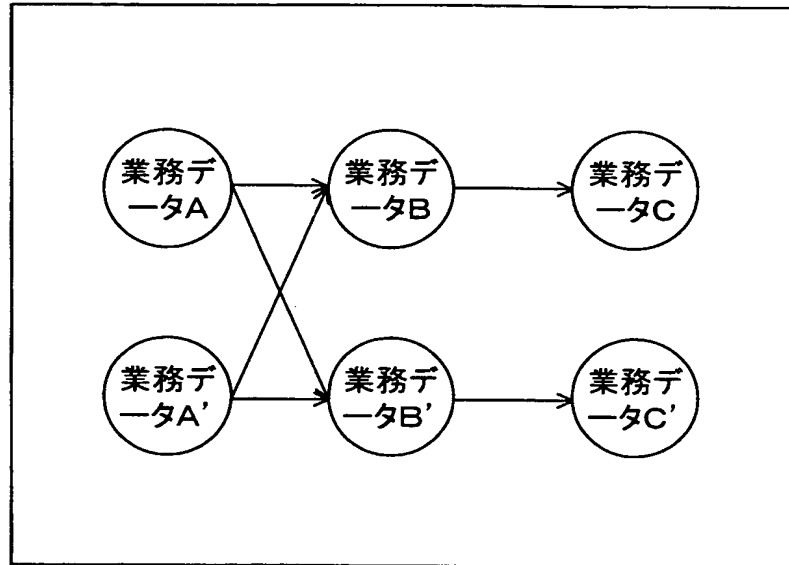
```
<bpm:event xmlns:bpm="urn:fj-bpm-event">
  <bpm:modelName>製品プロセス</bpm:modelName>
  <bpm:eventData>
    <bpm:eventName>受注</bpm:eventName>
    <bpm:startTime>20030605T103500</bpm:startTime>
  </bpm:eventData>
  <bpm:primaryKey type="受注伝票">o001</bpm:primaryKey>
  <bpm:property name="受注金額">1000</bpm:property>
  <bpm:property name="納品希望日">2003/6/20</bpm:property>
</bpm:event>
```

(B)

発注イベントデータ

```
<bpm:event xmlns:bpm="urn:fj-bpm-event">
  <bpm:modelName>製品プロセス</bpm:modelName>
  <bpm:eventData>
    <bpm:eventName>発注</bpm:eventName>
    <bpm:startTime>20030605T103500</bpm:startTime>
  </bpm:eventData>
  <bpm:primaryKey type="発注伝票">p101</bpm:primaryKey>
  <bpm:relativeKey type="受注伝票">o001</bpm:relativeKey>
  <bpm:property name="製番">12345</bpm:property>
  <bpm:property name="個数">100</bpm:property>
  <bpm:property name="単価">1000</bpm:property>
</bpm:event>
```


【図 1 2】



【図 13】

フロー表示用SVGファイル

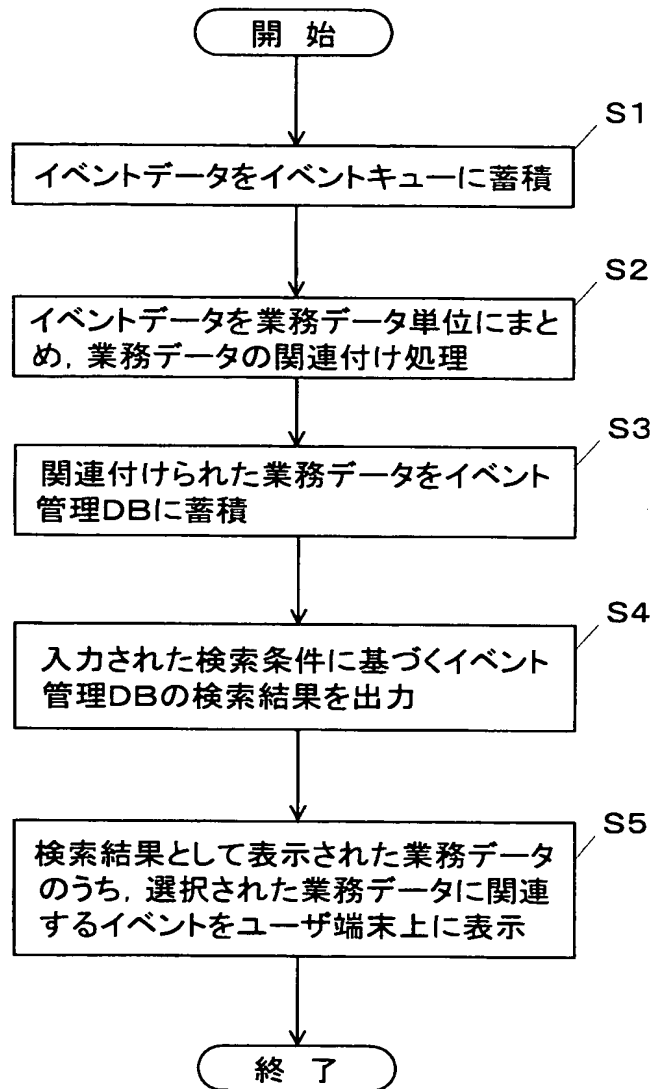
```

<?xml version="1.0" encoding="UTF-8"?>
<svg xmlns:bpm="urn:fujitsu-bpm" id="bpm" width="1333" height="1000">
  <defs>
    <marker id="arrowhead" viewBox="0 0 10 10" refX="10" refY="5" markerUnits="strokeWidth" markerWidth="4"
      markerHeight="3" orient="auto">
      <path d="M 0 0 L 10 5 L 0 10 z" />
    </marker>
  </defs>
  <g>
    <circle bpm:id="z+wPK9Hwsk6anJ9hz5R7zg" cx="143" cy="51" r="10" style="fill:#6FC498" stroke="#666666" stroke-width="2" />
    <text bpm:id="z+wPK9Hwsk6anJ9hz5R7zg" bpm:text="1" x="168" y="53"></text>
  </g>
  <rect bpm:id="受注" x="91" y="104" width="100" height="100" style="fill:#CCCCDD" stroke="none" />
  <text bpm:id="受注" x="141" y="132" text-anchor="middle">受注</text>
  <text bpm:id="受注" bpm:text="0" x="141" y="144" text-anchor="middle"></text>
  <text bpm:id="受注" bpm:text="1" x="193" y="147"></text>
  <text bpm:id="受注" bpm:text="2" x="193" y="126"></text>
  <text bpm:id="受注" bpm:text="3" x="193" y="106"></text>
  <rect bpm:id="受注" x="91" y="104" width="100" height="100" style="fill:#808285" stroke="none" />
  <text bpm:id="受注" x="141" y="116" fill="white" text-anchor="middle">&lt;&lt;販売&gt;&gt;</text>
</g>
</svg>

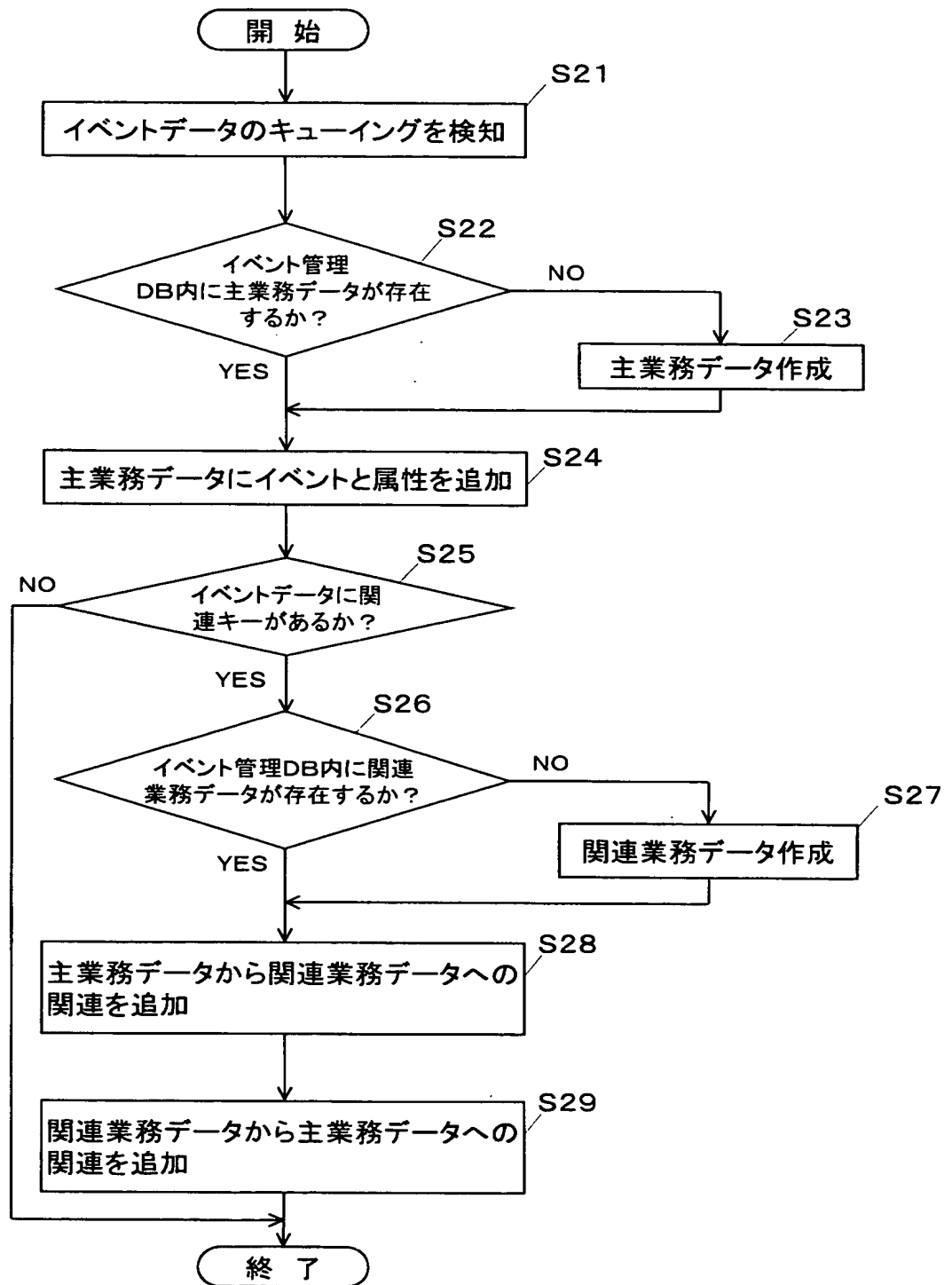
```

ベクターデータ内に識別IDを埋
め込んだ形

【図 14】



【図 15】



【図 1 6】

(A)

イベントデータ: 発注

```
<bpm:event xmlns:bpm="urn:fj-bpm-event">
  <bpm:modelName>製品プロセス</bpm:modelName>
  <bpm:eventData>
    <bpm:eventName>発注</bpm:eventName>
    <bpm:startTime>20030827T1000</bpm:startTime>
  </bpm:eventData>
  <bpm:primaryKey type="発注伝票">p101</bpm:primaryKey>
  <bpm:relativeKey type="受注伝票">o001</bpm:relativeKey>
  <bpm:property name="単価">1000</bpm:property>
</bpm:event>
```

(B)

主業務データ: 発注伝票

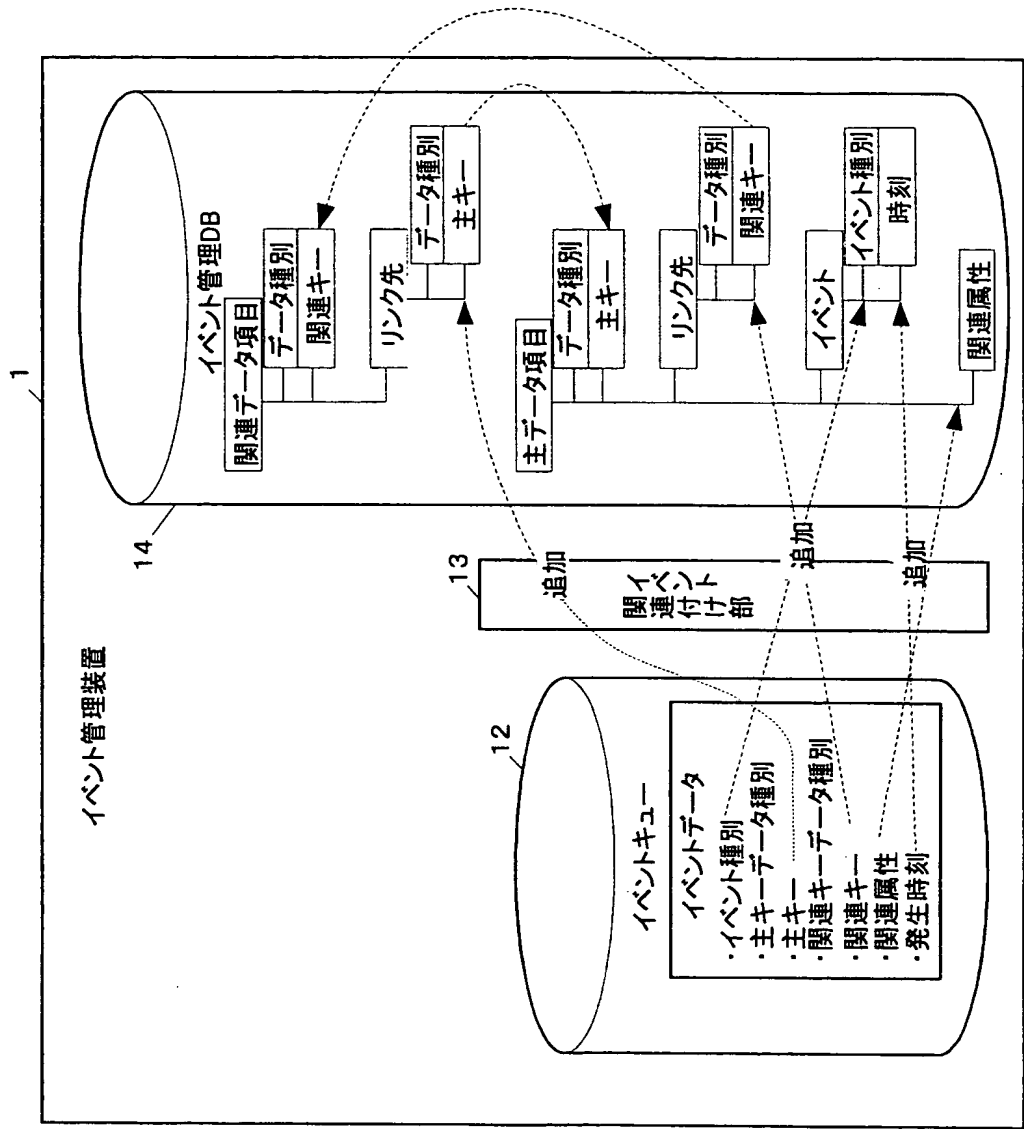
```
<bpm:businessData xmlns:bpm="urn:fj-bpm-event">
  <bpm:modelName>製品プロセス</bpm:modelName>
  <bpm:primaryKey type="発注伝票">p101</bpm:primaryKey>
  <bpm:relativeKey type="受注伝票">o001</bpm:relativeKey>
  <bpm:eventData>
    <bpm:eventName>発注</bpm:eventName>
    <bpm:startTime>20030827T1000</bpm:startTime>
  </bpm:eventData>
  <bpm:property name="単価">1000</bpm:property>
</bpm:businessData>
```

(C)

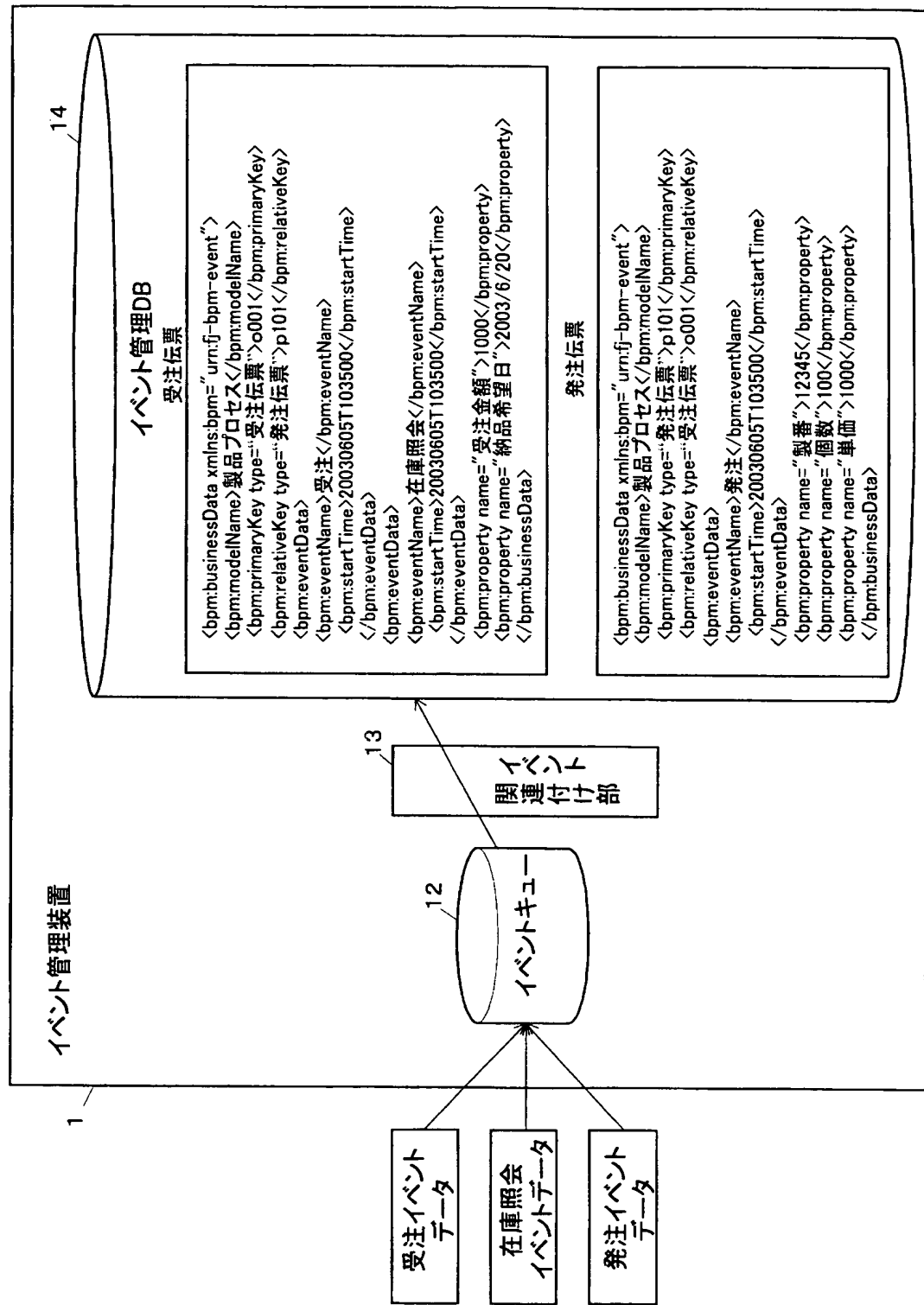
関連業務データ: 受注伝票

```
<bpm:businessData xmlns:bpm="urn:fj-bpm-event">
  <bpm:modelName>製品プロセス</bpm:modelName>
  <bpm:primaryKey type="受注伝票">o001</bpm:primaryKey>
  <bpm:relativeKey type="発注伝票">p101</bpm:relativeKey>
</bpm:businessData>
```

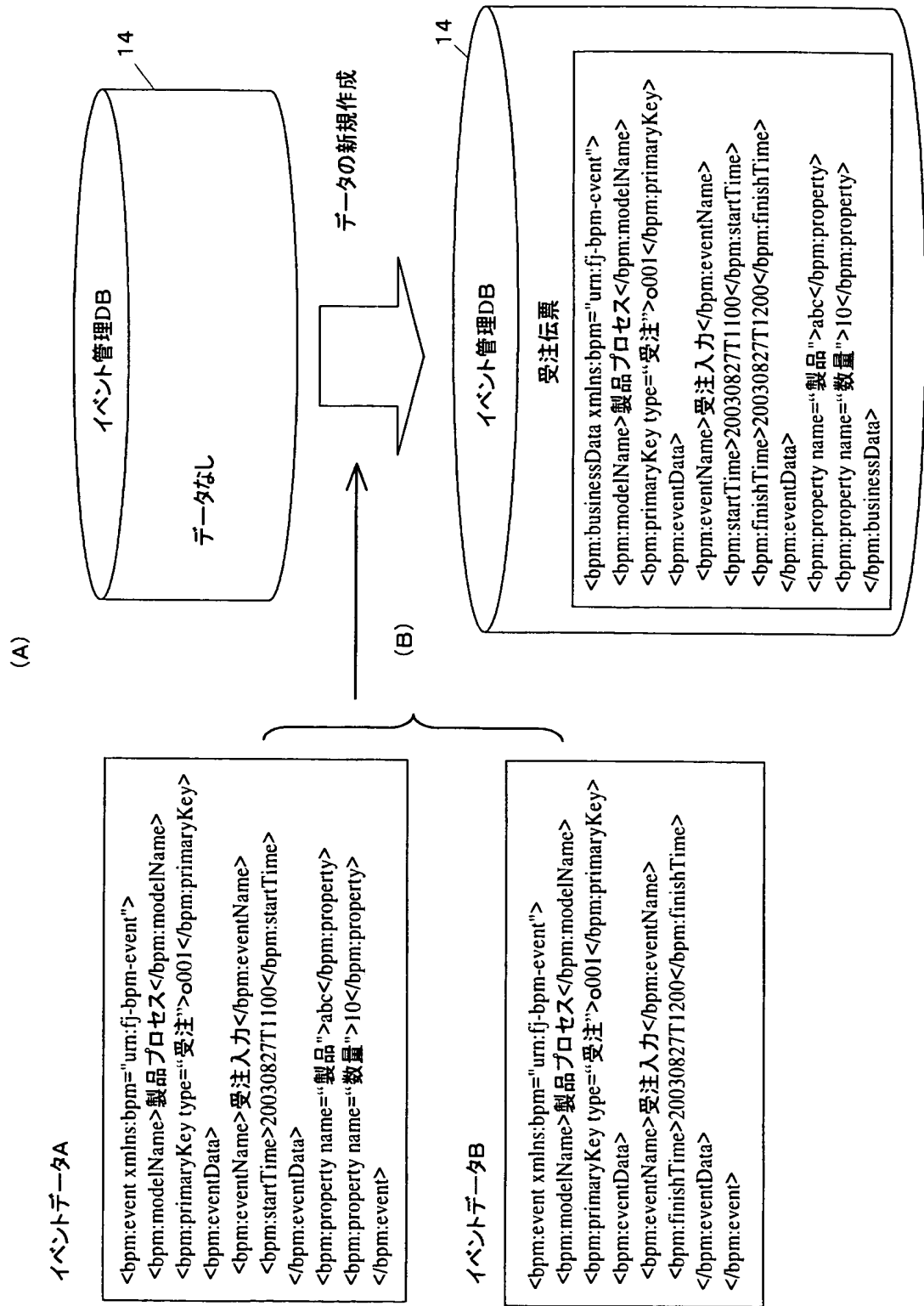
【図 17】



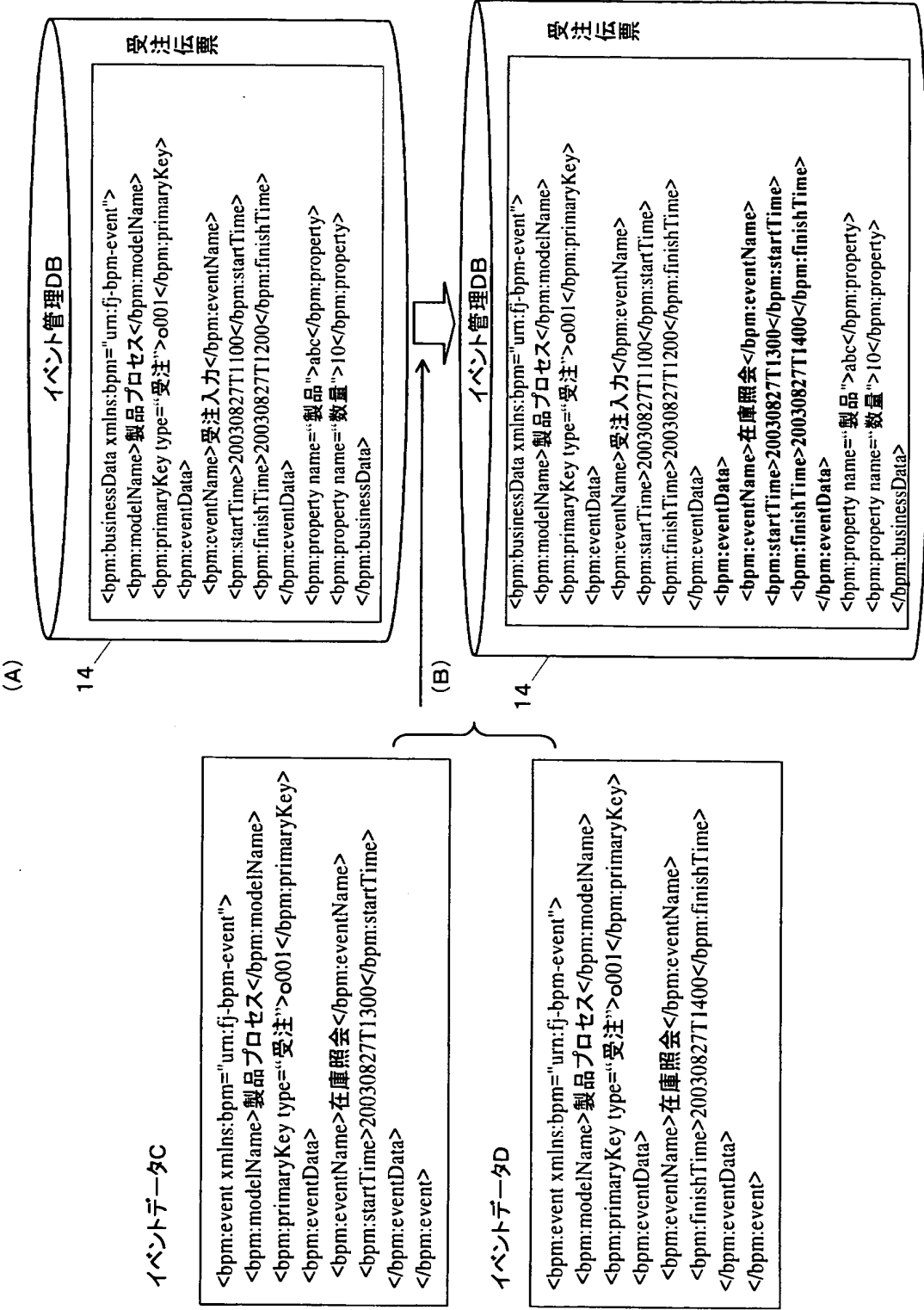
【図 18】



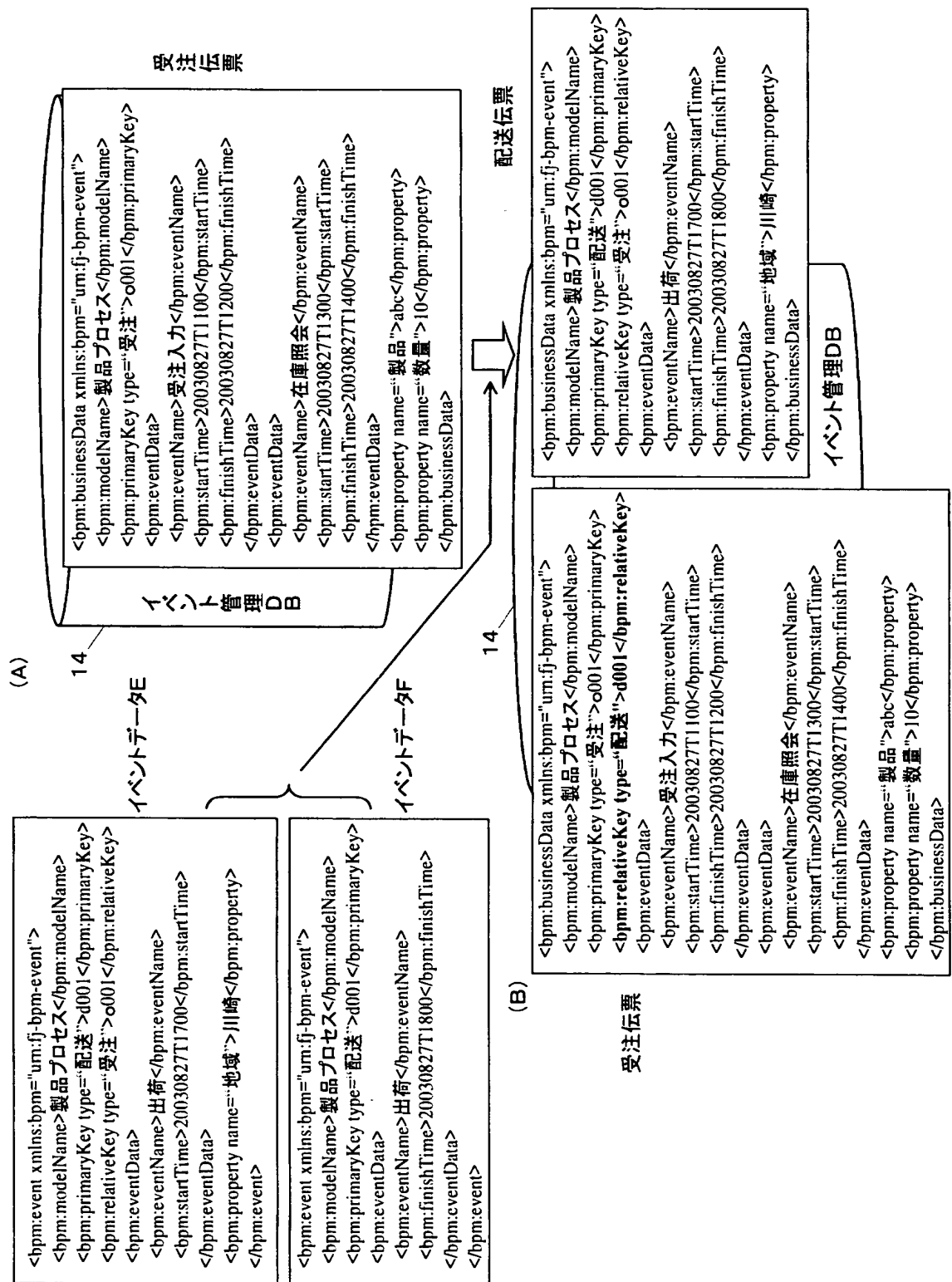
【図 19】



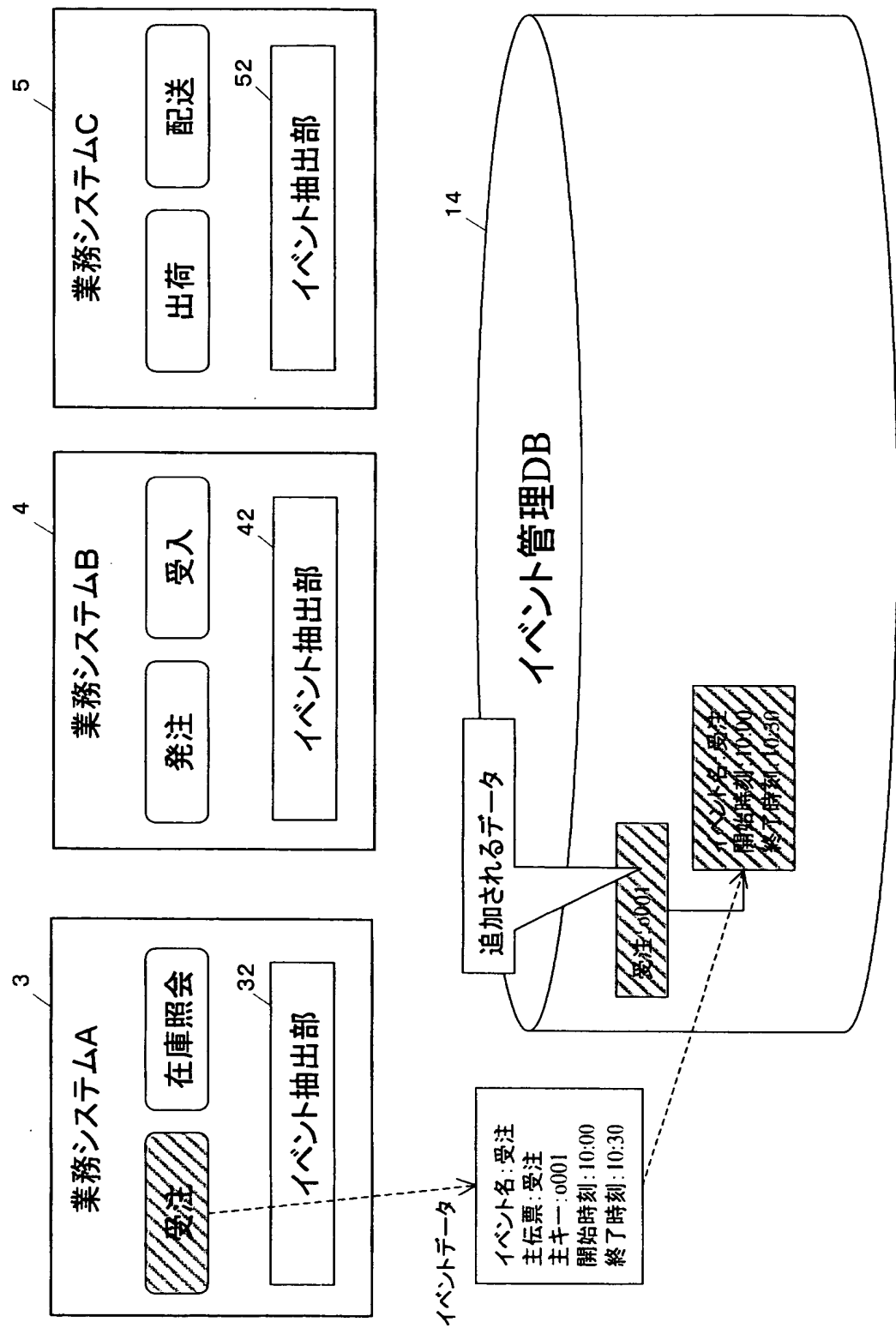
【図 20】



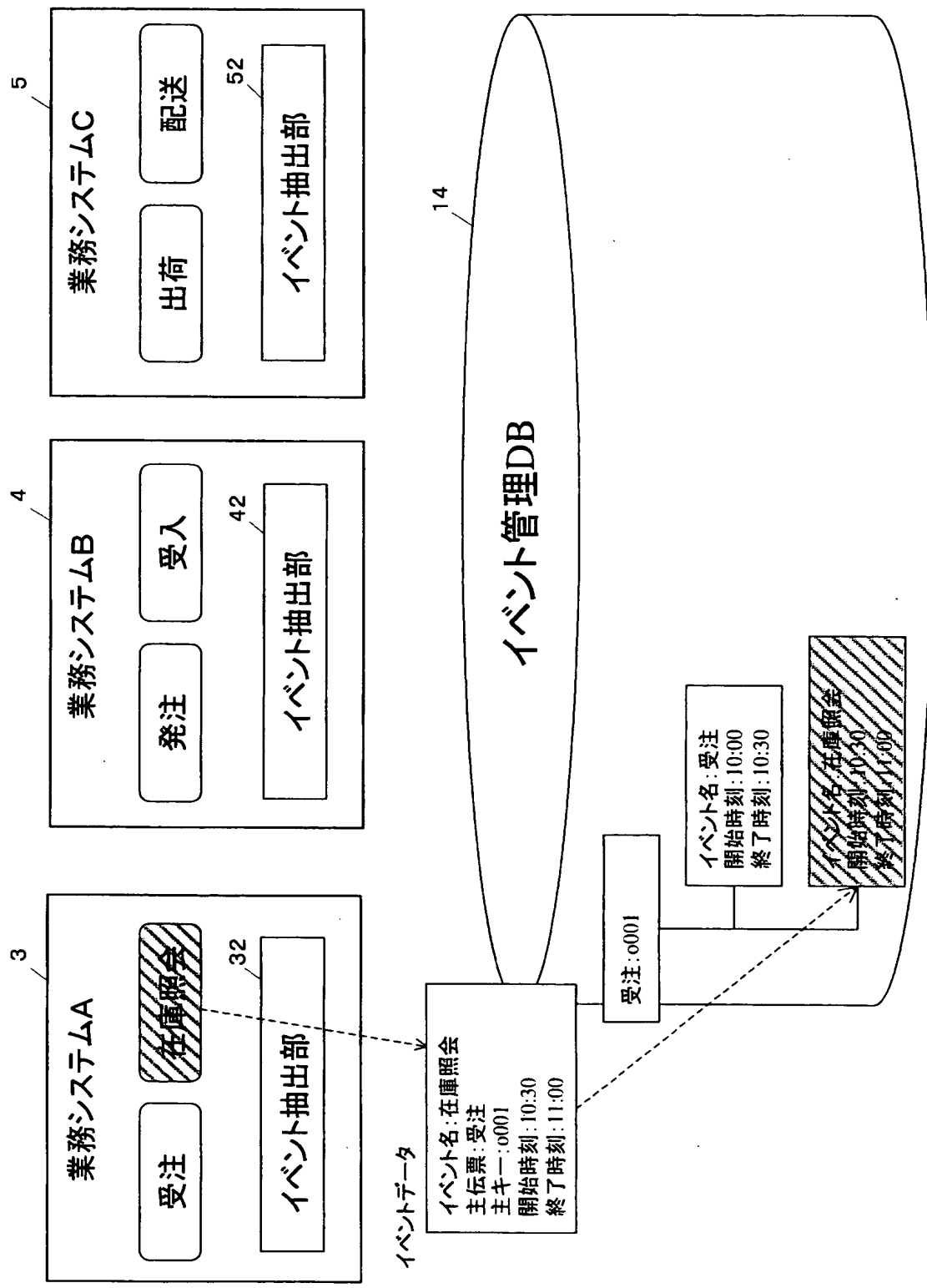
【図 21】



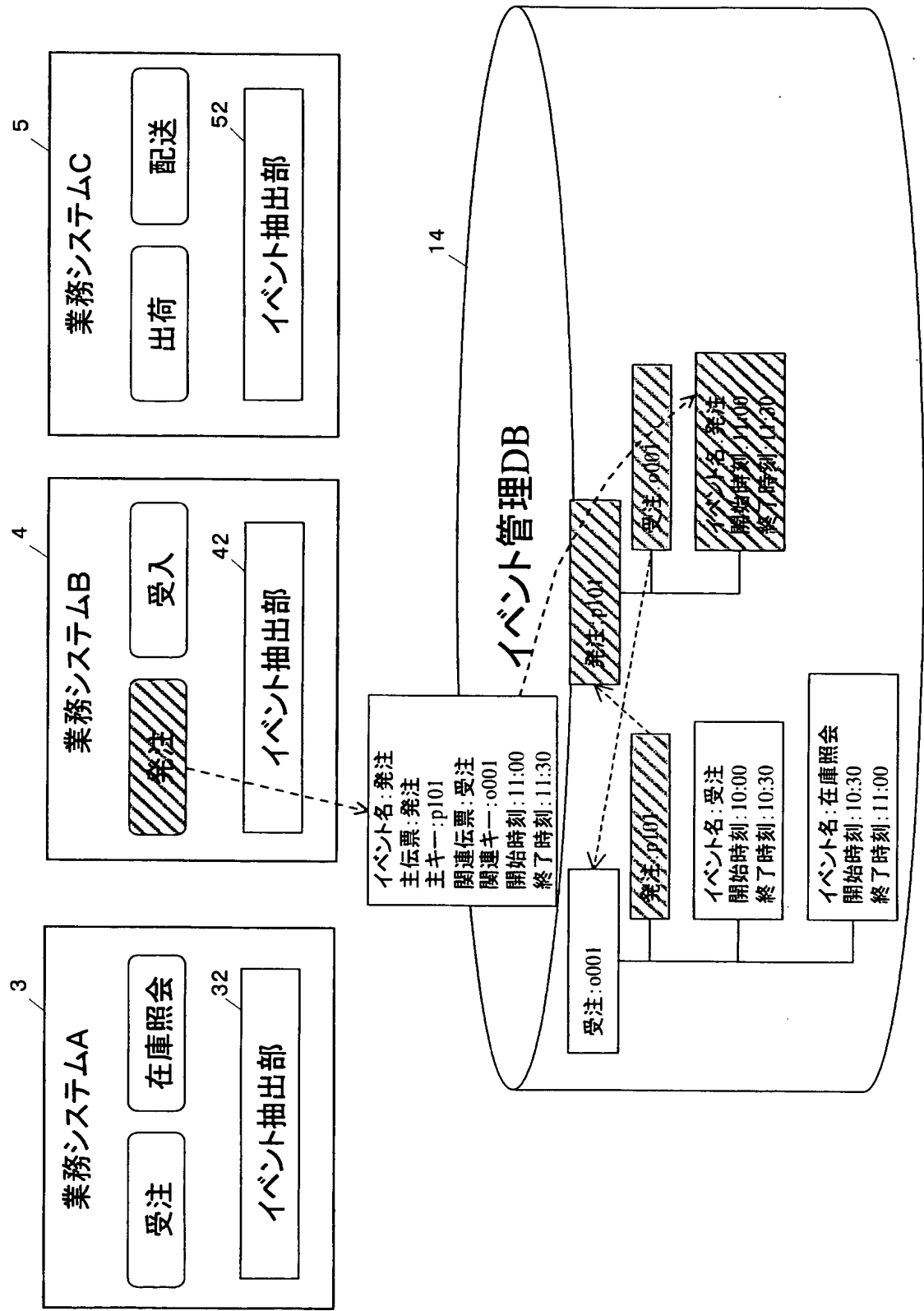
【図 22】



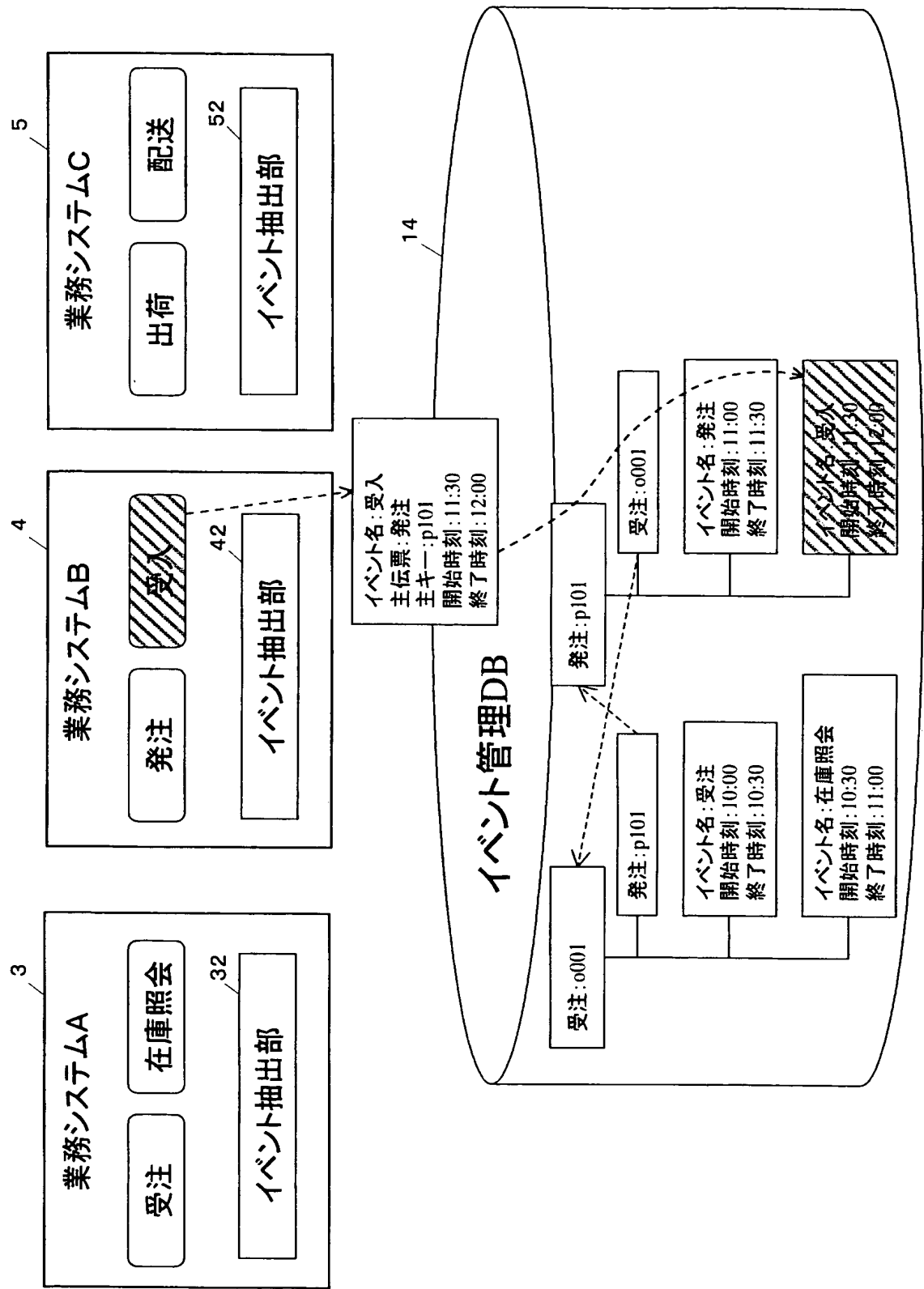
【図 23】



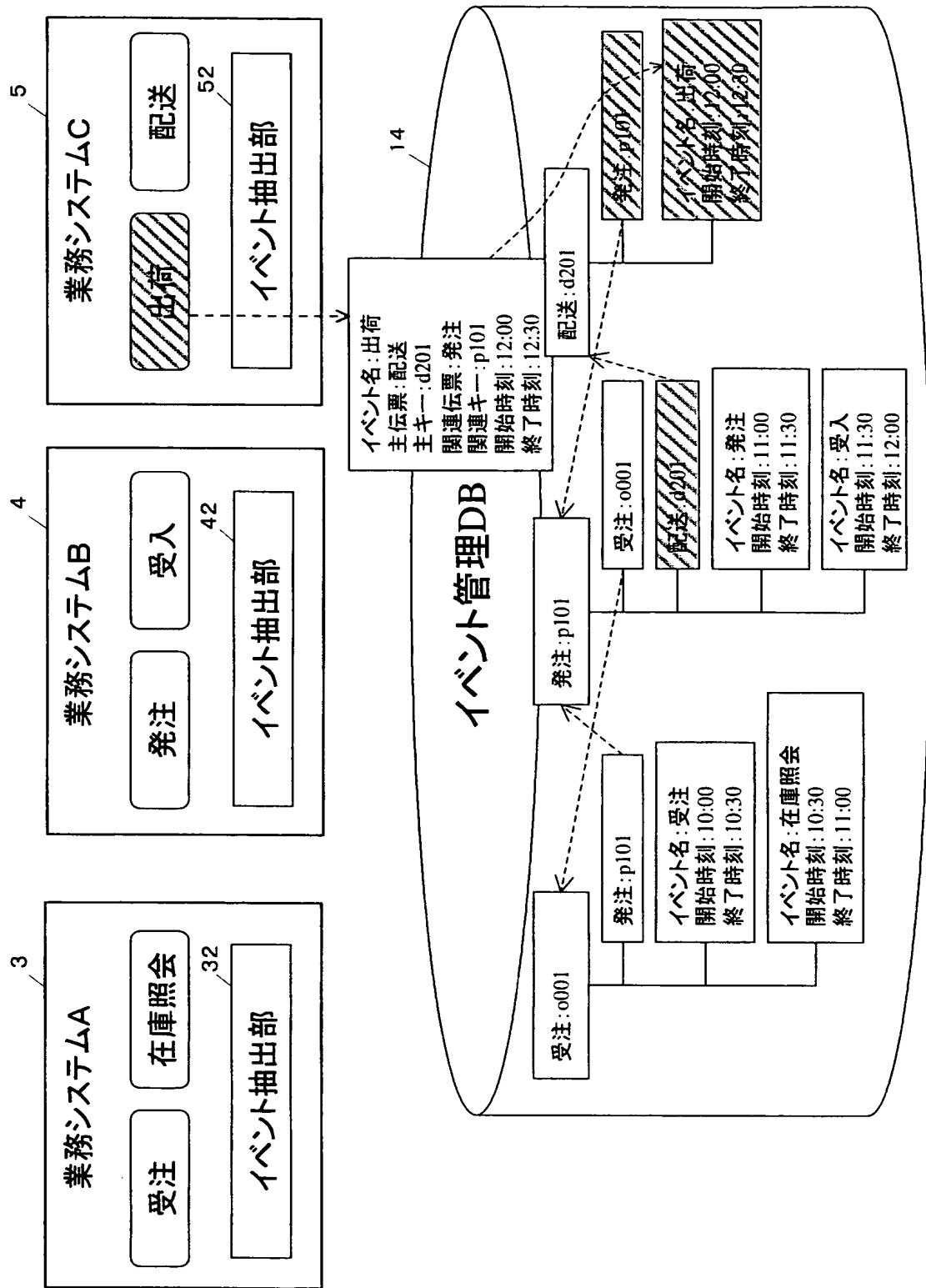
【図 24】



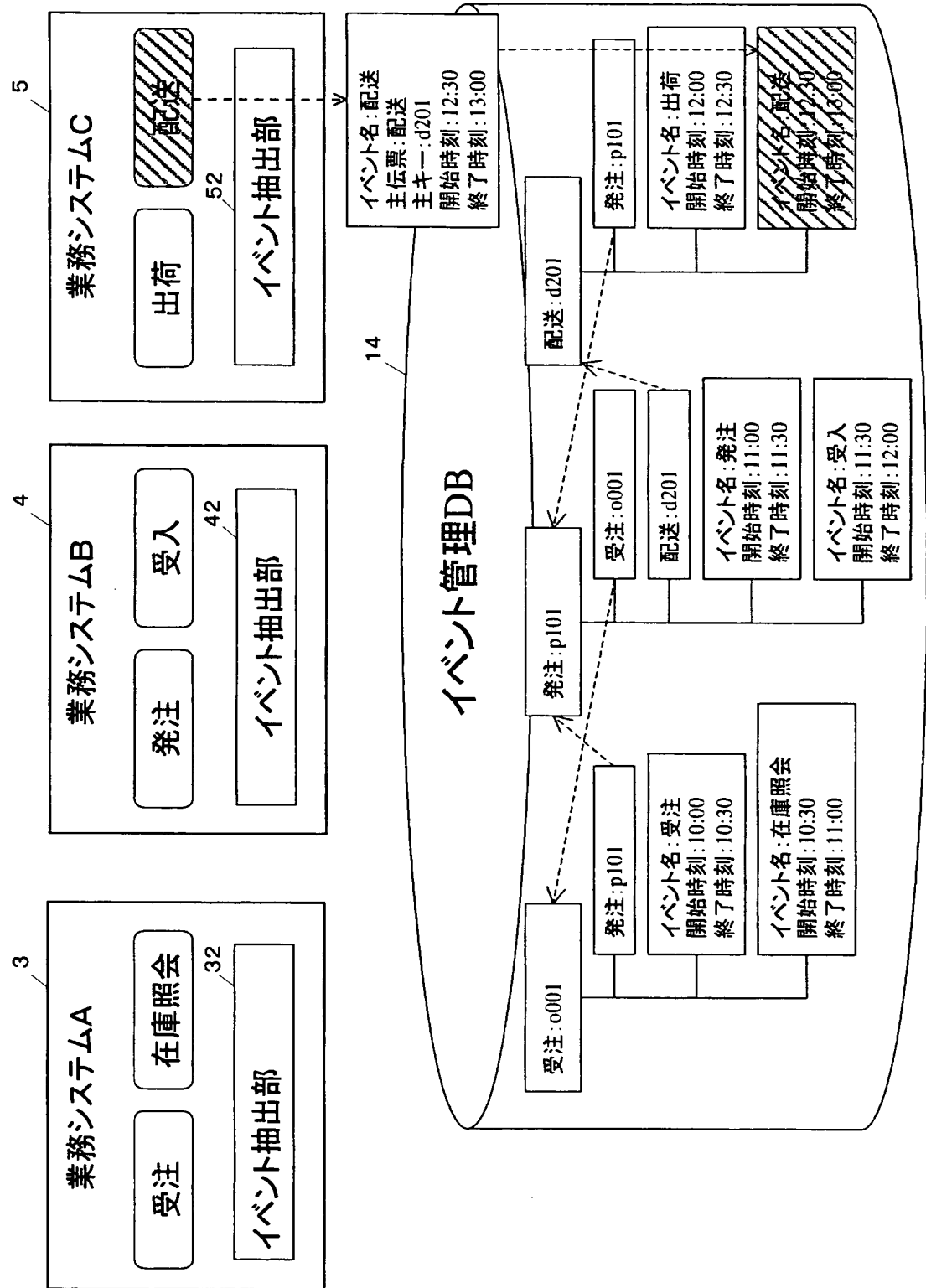
【図 25】



【図 2 6】



【図 27】



【図 28】

—

□

×

検索条件入力 — 業務プロセスラッキング

業務プロセス: sample1 ▼

検索対象: 配送 ▼

検索項目1: 伝票番号 ▼ 値:

検索項目2: 項目を選択してください ▼ 値:

検索項目3: 項目を選択してください ▼ 値:

検索範囲: 2003 ▼ 年 1 ▼ 月 1 ▼ 日 ~ 2003年 ▼ 年 10 ▼ 月 1 ▼ 日

結果表示方法: ◎ イベントエクスプローラで表示 ○ 一覧で表示

検索実行 クリア 戻る ヘルプ

【図 2 9】

検索条件入力 ― 業務プロセスラッキング

業務プロセス:

検索対象:

検索項目 1:	<input type="button" value="項目を選択してください"/>	値:	<input type="text"/>
検索項目 2:	<input type="button" value="項目を選択してください"/>	値:	<input type="text"/>
検索項目 3:	<input type="button" value="項目を選択してください"/>	値:	<input type="text"/>

検索範囲: 2003 年 1 月 1 日 ~ 2003 年 1 月 1 日

結果表示方法: ○ イベントエクスペローラで表示 ◎ 一覧で表示

【図 3 0】

伝票の検索

検索条件

プロセス：製品販売

検索対象：受注

検索範囲：2003年1月1日～2003年1月1日

検索結果

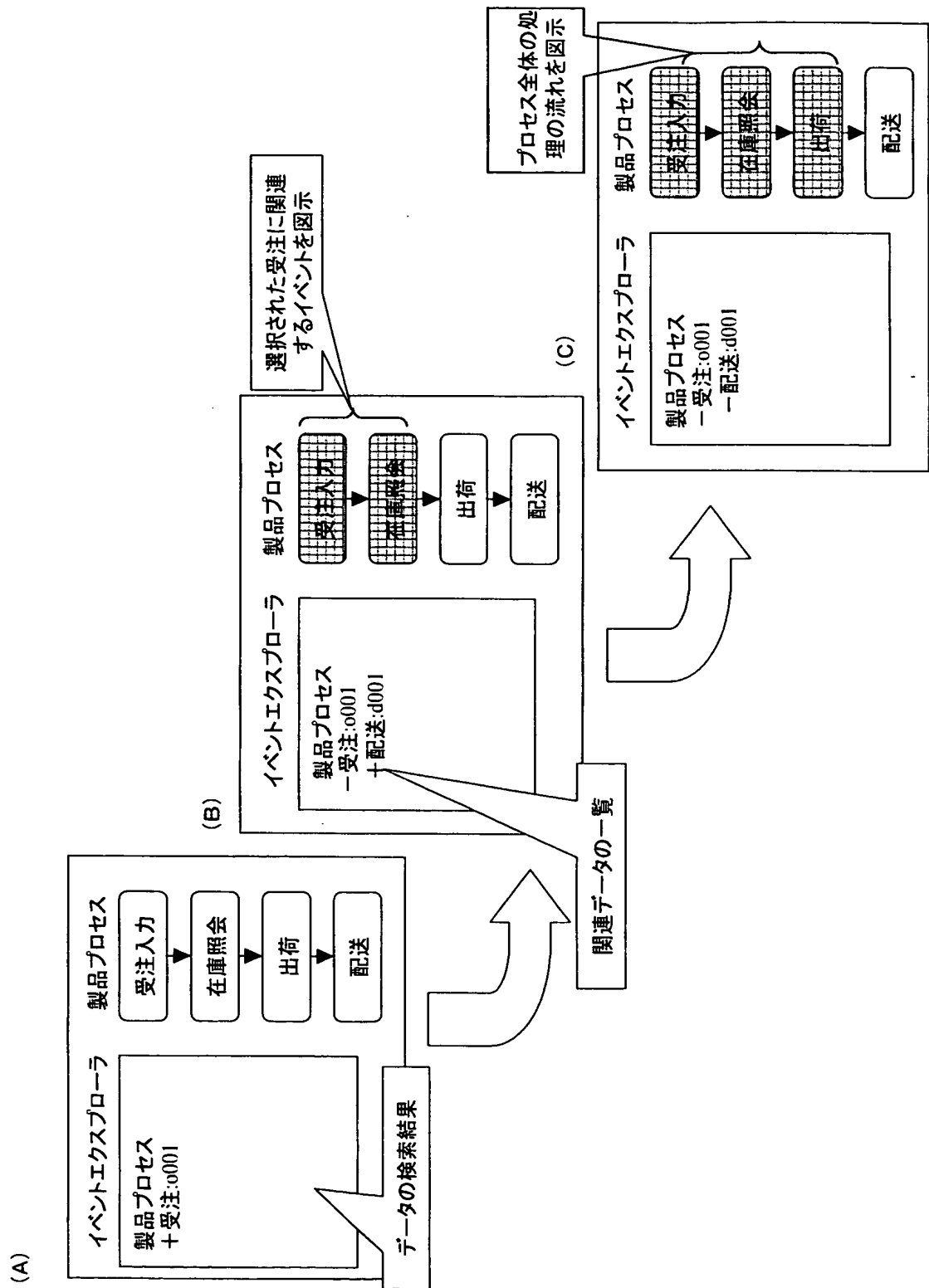
伝票番号：o002

伝票番号：o001

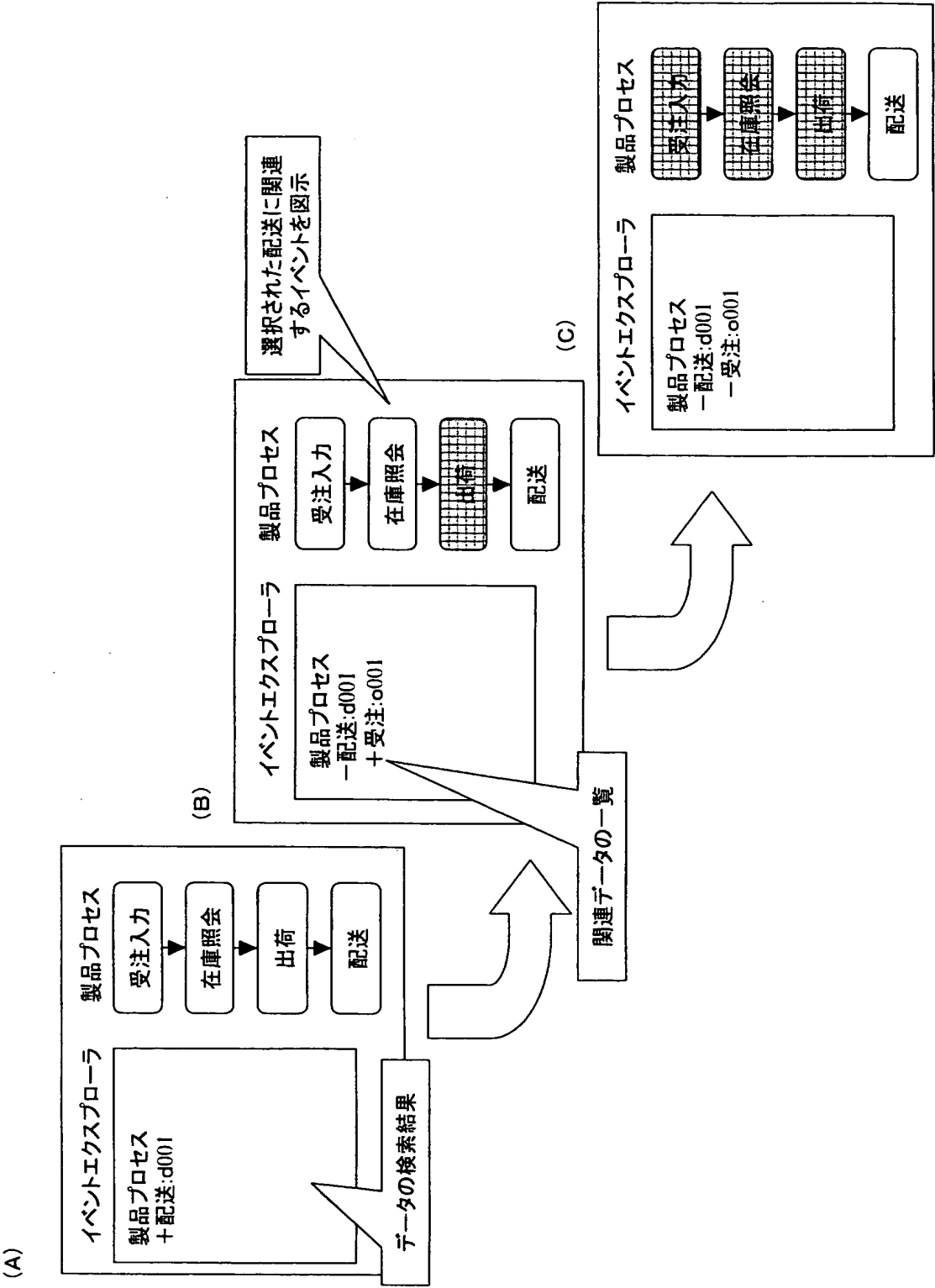
伝票番号：o011

検索条件入力

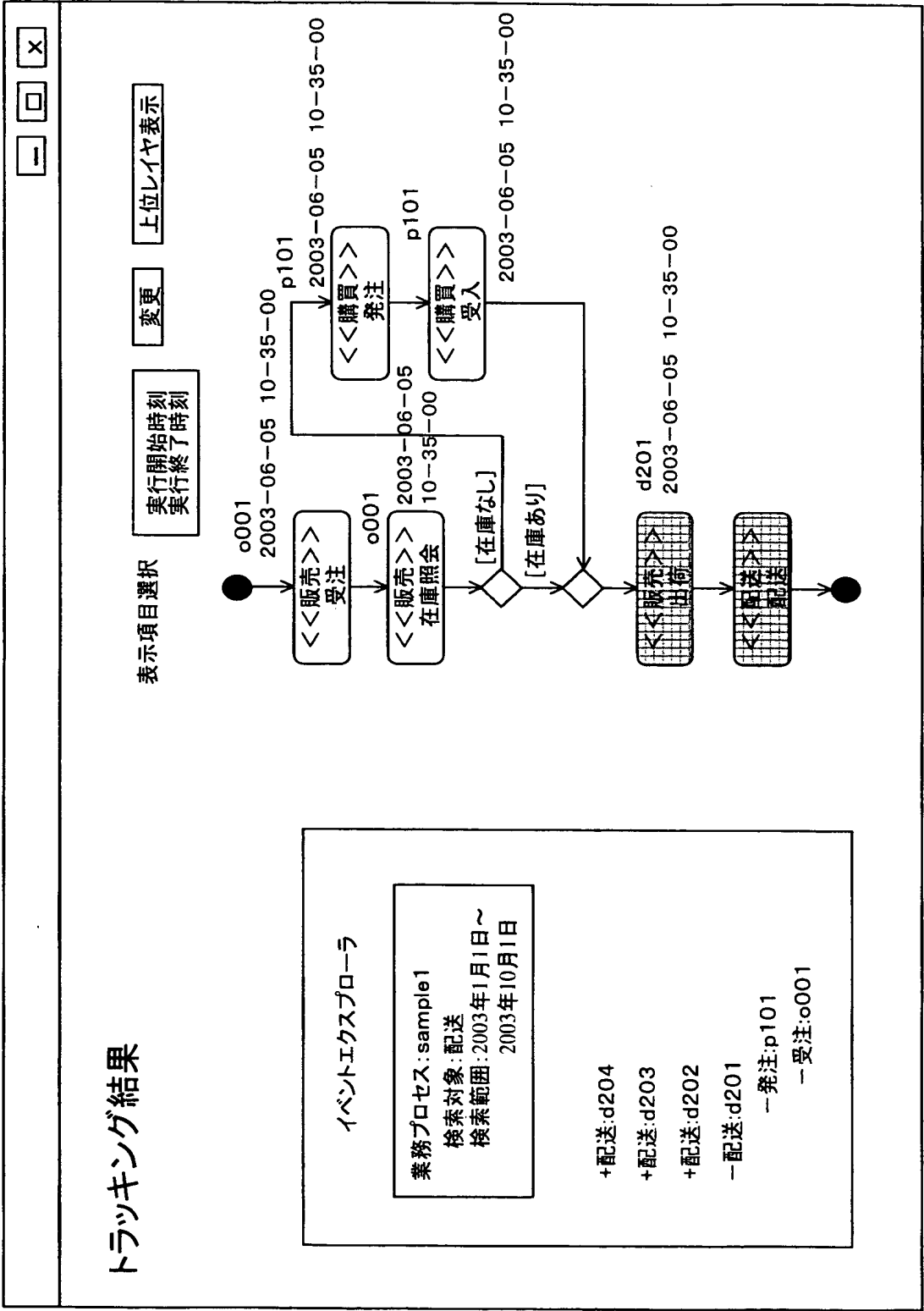
【図 31】



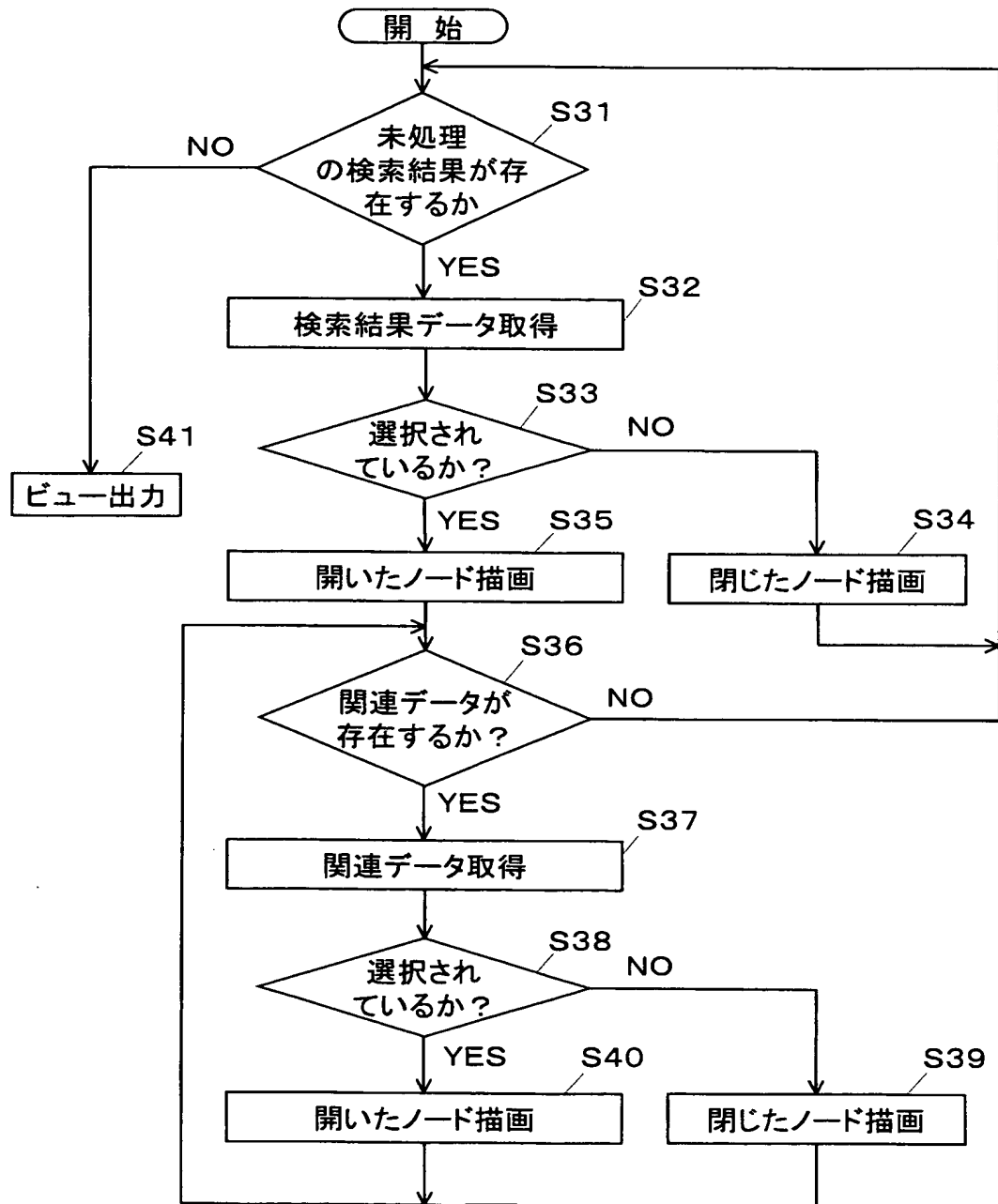
【図 32】



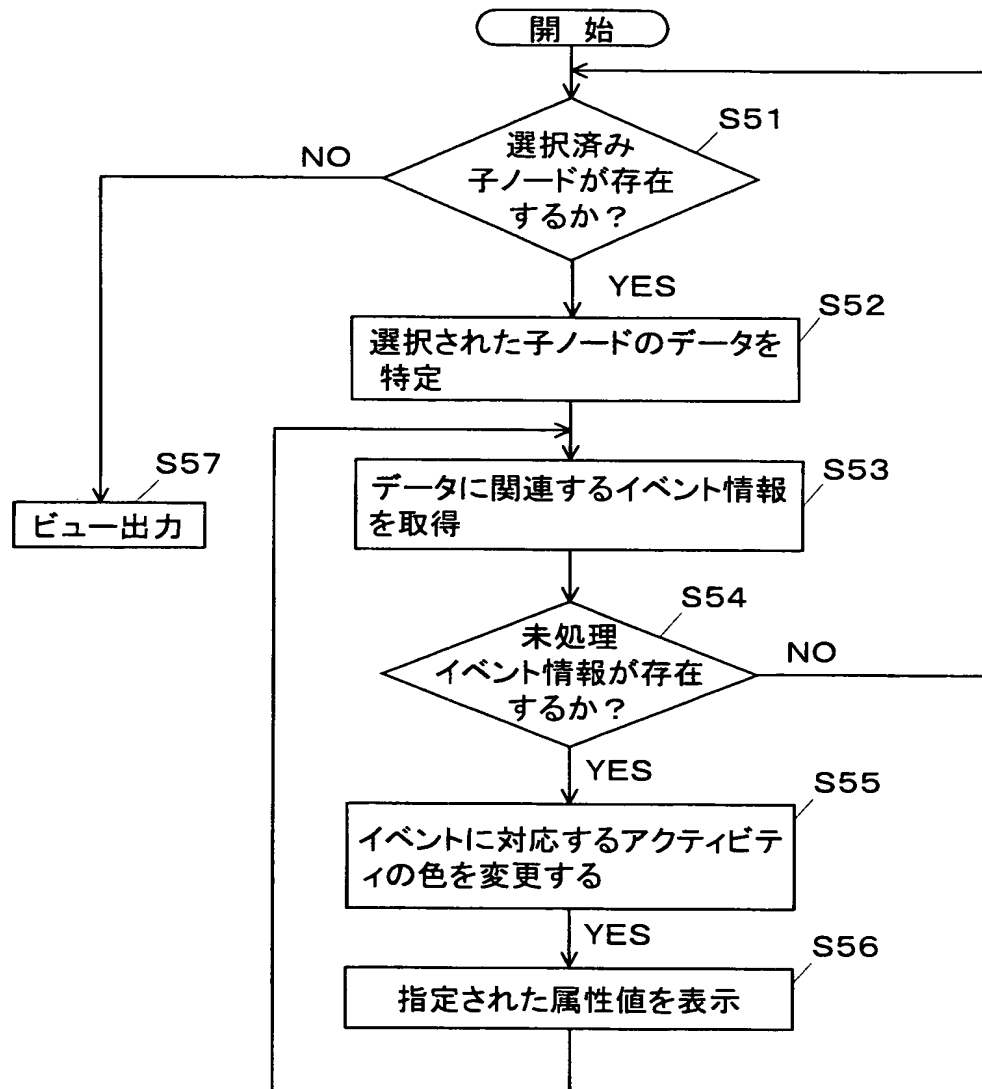
【図 33】



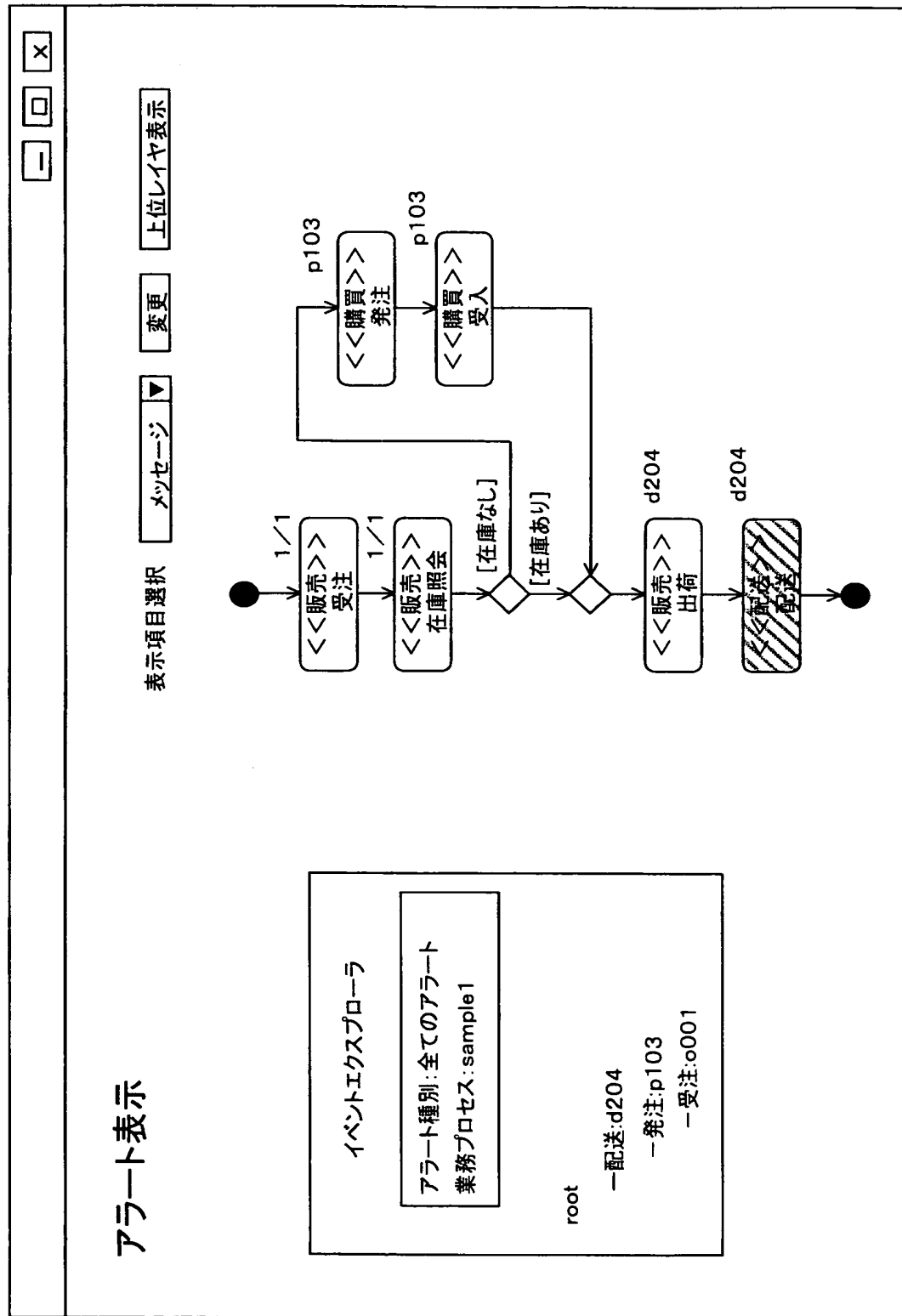
【図 34】



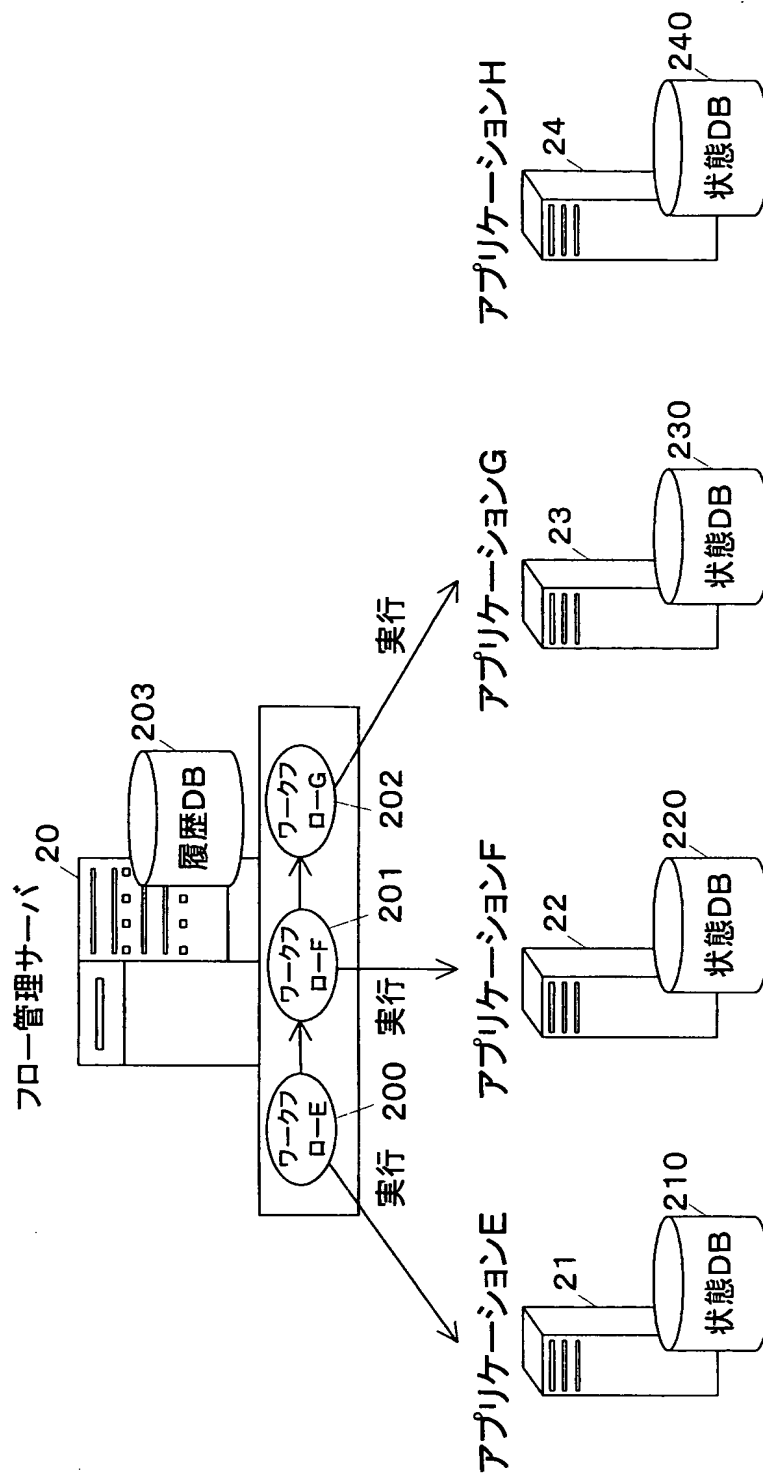
【図 35】



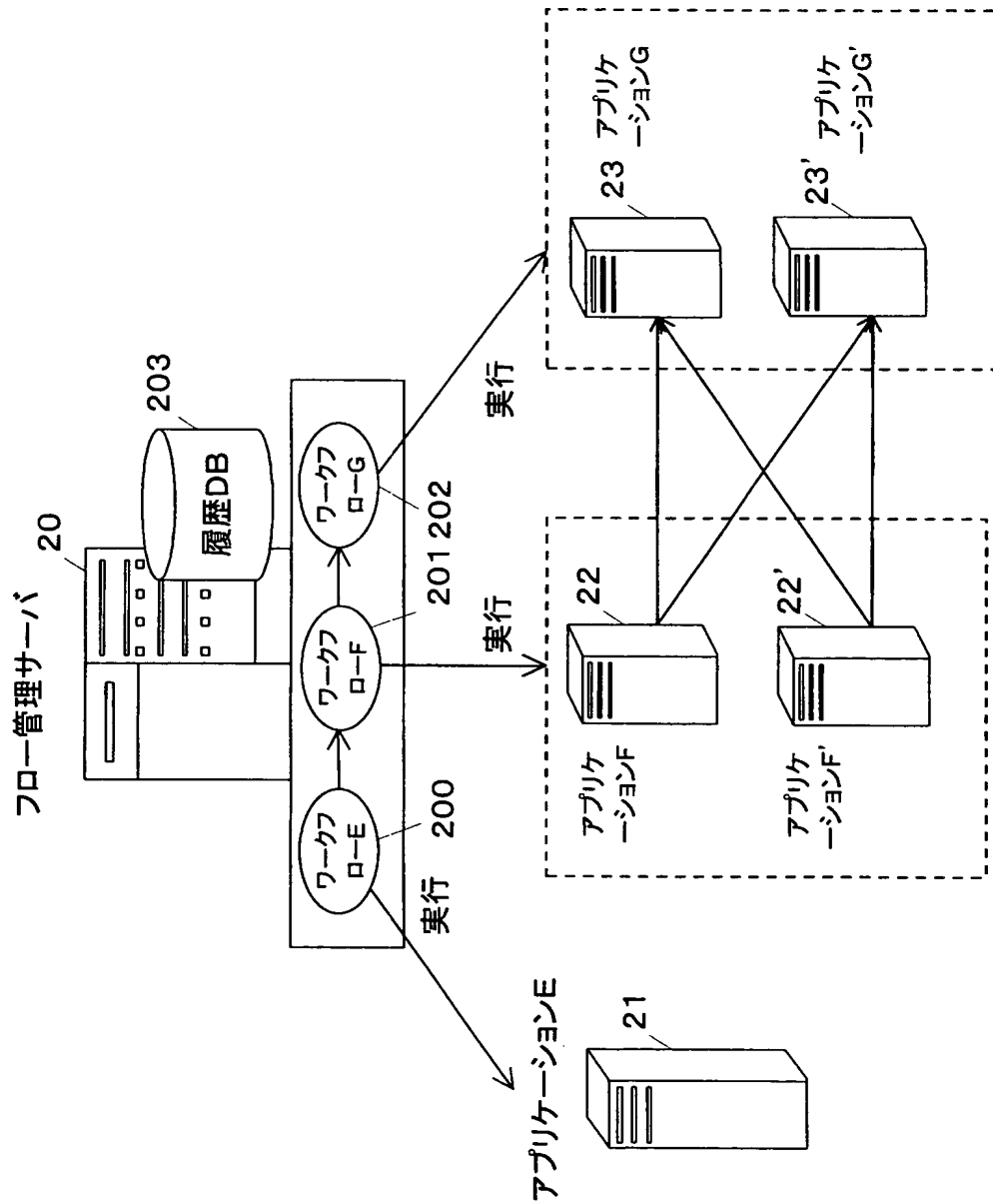
【図 36】



【図 37】



【図 38】



**【書類名】 要約書****【要約】**

【課題】 業務プロセストラッキング装置に関し、異なる業務システム間を跨いで行われる複数のアプリケーションからなる業務プロセスの流れを、既存システムを変更することなく追跡することを可能とすることを目的とする。

【解決手段】 イベント管理装置 1 は、各業務システム A 3 ～ C 5 のイベント抽出部 3 2 ～ 5 2 がイベント抽出定義に基づいて抽出したイベントデータを収集し、イベントキュー 1 2 にキューイングする。イベント関連付け部 1 3 は、イベントデータを業務データ単位にまとめ、業務データ間の関連付けを行ってイベント管理 DB 1 4 に蓄積する。ユーザ端末 6 から検索条件が入力されると、出力部 1 6 が検索条件に従ってイベント管理 DB 1 4 を検索し、業務データ間の関連をツリー形式でユーザ端末 6 に出力し表示する。

【選択図】 図 1

特願 2 0 0 3 - 3 4 6 2 7 1

出 願 人 履 歴 情 報

識別番号 [0 0 0 0 0 5 2 2 3]

1. 変更年月日 1 9 9 6 年 3 月 2 6 日

[変更理由] 住所変更

住 所 神奈川県川崎市中原区上小田中 4 丁目 1 番 1 号

氏 名 富士通株式会社